



Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

M-Eco enhanced Adaptation Service (D5.2)

Dolog, Peter; Durao, Frederico; Lage, Ricardo Gomes; Leginus, Martin; Pan, Rong

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Dolog, P., Durao, F., Lage, R. G., Leginus, M., & Pan, R. (2012). *M-Eco enhanced Adaptation Service (D5.2)*.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

M-Eco enhanced Adaptation Service (D5.2)

Project title:	Medical Ecosystem - Personalized Event-Based Surveillance
Project acronym:	M-Eco
Project number:	247829
Project instrument:	EU FP7 Small or Medium-scale Focused Research Projects (STREP)
Project thematic priority:	Information and Communication Technologies (ICT)
Document type:	D (deliverable)
Nature of document:	R/P (report and prototype)
Responsible editors:	Peter Dolog
Authors:	Peter Dolog, Frederico Durao, Ricardo Lage, Martin Leginus, Rong Pan
Quality Assessor:	Göran Kirchner Email:KirchnerG@rki.de
Contributing participants:	AAU
Contributing workpackages:	WP5

Abstract

In this report, we present the improvements of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. This component is focused on four main areas of interest to users of surveillance systems: presentation options for recommendation and adaptation, user and group models, user classification and modeling algorithms, and recommendation, adaptation and personalization strategies. In each of these areas, we propose new methods and improvements over those presented in the previous deliverable. For example, we extend the user profiling with the user's stated preferences for a set of locations and we propose new approaches for recommendations to groups and navigation through documents using tags. The extensions and new methods address diverse aspects of the M-Eco system such as motivation scenarios, configuration management, formal models, implementations and some real use case scenarios on how personalization can be applied to support medical surveillance. The result of personalization can be seen, for example, as a reordering on the ranking of recommendations or search results or a personalized tag cloud leading users to quick access the information fulfilling the users interests. The main contributions of the report are spatial reasoning methods evaluated in group recommendations, personalized tag clouds, and web services integration.

Keyword List

social tagging, recommender system, tag cloud, group recommendation

M-Eco enhanced Adaptation Service (D5.2)

Peter Dolog, Frederico Durao, Ricardo Lage, Martin Leginus, Rong Pan

Department of Computer Science, Aalborg University
Email:dolog,fred,ricardol,mleginus,rpan@cs.aau.dk

10 January 2012

Abstract

In this report, we present the improvements of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. This component is focused on four main areas of interest to users of surveillance systems: presentation options for recommendation and adaptation, user and group models, user classification and modeling algorithms, and recommendation, adaptation and personalization strategies. In each of these areas, we propose new methods and improvements over those presented in the previous deliverable. For example, we extend the user profiling with the user's stated preferences for a set of locations and we propose new approaches for recommendations to groups and navigation through documents using tags. The extensions and new methods address diverse aspects of the M-Eco system such as motivation scenarios, configuration management, formal models, implementations and some real use case scenarios on how personalization can be applied to support medical surveillance. The result of personalization can be seen, for example, as a reordering on the ranking of recommendations or search results or a personalized tag cloud leading users to quick access the information fulfilling the users interests. The main contributions of the report are spatial reasoning methods evaluated in group recommendations, personalized tag clouds, and web services integration.

Keyword List

social tagging, recommender system, tag cloud, group recommendation

Contents

1	Introduction	1
2	Motivation	3
2.1	Advantages and Motivations for the Personalization and Adaptation Methods . .	4
2.2	User Requirements Addressed	5
2.3	Personalization Scenario	5
2.4	Signal and document recommendation through M-Eco portal	9
2.5	Modeling group work interactions	9
2.6	Navigating through documents for validation of signals	11
3	Design and Architecture	12
3.1	Component Integration	12
3.2	Component Packaging	14
4	Adaptive Tuning and Personalization Models	15
4.1	Analysis of Presentation Options for Recommendation and Adaptation	16
4.1.1	User Feedback Options	19
4.2	User and User Group Model Definition	20
4.2.1	Tagging Model	20
4.2.2	Group Modeling	21
4.2.3	Location Preference Model	22
4.3	User Classification and Modeling Algorithms	22
4.3.1	Semantic Enhanced Tag-Based Recommendation	23
4.3.2	Expanding Tags with Neighbors For Improving Search Retrieval	23
4.3.3	Spectral Clustering of Tag Neighbors for Recommendation	24
4.4	Recommendation, Adaptation and Personalization Strategies	26
4.4.1	Recommendations based on Signal Definitions	26
4.4.2	Tag-Based Recommendations	26
4.4.3	Recommendations based on Locations	26
4.4.4	Group Recommendations	29
4.4.5	Personalized Tag Cloud Adaptation and Navigation	31
4.4.6	Generative Model of User Taggings	35
5	Evaluation	36
5.1	User Evaluation of Recommendation Presentation and Feedback Options	36
5.1.1	Recommendation Results	38
5.1.2	Group Discussion	39
5.2	Evaluation of Spectral Clustering of Tag Neighbors for Recommendation	40
5.2.1	Precision Evaluation and Evaluation Results	41
5.3	Personalized Tag Cloud Evaluation	42
5.4	Evaluation of Generative Model of User Taggings	45
5.5	User Evaluation of Group Recommendations	46
5.6	Evaluation of Location Recommendations in a Group Setting	52
5.7	Stress tests	53
5.7.1	Tag cloud stress tests	55

6	Deployment	56
6.1	Installation Requirements	56
6.2	Web Services	57
6.2.1	Web Service Providers	58
6.2.2	Web Service Consumers	58
7	Related work	58
7.1	User Modeling	58
7.2	Recommendations for Health Care	59
7.3	Group Recommendations	60
7.4	Tag Expansion in Recommendation	61
7.5	Personalized Tag cloud	61
7.6	Spatial Reasoning	62
8	Conclusion and Future Works	63

1 Introduction

This deliverable presents enhanced methods of user and group modeling, and recommendation for event detection, which are part of the Adaptive Tuning and Personalization (WP5) component of the M-Eco system. It aims at adapting the information presented in the system based on what is likely to be more useful to a particular user or a group of users in different scenarios. In general, such aim is directed to users who are faced with a potentially overwhelming list of items to evaluate. **Item**, in this case, is the general term used to represent anything that can be presented to a user, such as news articles, reports, blog posts as generated by methods from WP3 or signals in aggregated form as generated by methods from WP4. Adaptation is achieved through a process of modeling user interests in terms of items of the system, and recommending those that are more relevant.

In other words, the WP5 component adapts the M-Eco system to the needs of the user. It is an essential component in assisting users who have to synthesize an increasing number of facts, assess risks and react early to public health threats. Personalization and adaptation in the M-Eco project aims at customizing the items shown to the individual users, based on their use of the system.

In M-Eco, items used for recommendation are documents, tags and signals. **Documents** can refer to any source of information on the web identified by WP3 [45] as potentially relevant for health surveillance. These include medical blog posts, news articles or videos discussing epidemics, and personal messages in social networks describing one or more symptoms. **Tags** are labels that summarize a document succinctly. They can be assigned by users of M-Eco, as a way to let them organize the documents presented to them, or generated automatically, as a way to help the users browse through the documents. Signals are aggregations of documents corresponding to a specific set of medical conditions (e.g., symptoms or diseases) and locations.

Signals are produced to specific user needs when matched with one or more of their signal definitions. These definitions are a set of rules specified by the user in terms of explicit medical conditions and locations he or she is interested in having the system monitoring. Once the user specifies signal definitions, this component can provide a toolset which mainly helps in the following way:

- in presenting and ordering (recommendations of) signals and documents as results of signal definition matchmaking; and,
- exploring and navigating users for validation purposes.

The toolset follows a general workflow which is depicted in Figure 1. The workflow consists of the tasks where the medical personnel: 1) defines a signal definition, 2) review of retrieval realized by validating recommended results which match the signal definition at the signal and document level and finally 3) issue of requests for actions if needed.

The M-Eco WP5 services are targeted at the support for the second task and, therefore, the figure depicts a link between those tasks and web page symbols which will provide services not only from WP5 to support the task. WP5 methods rely on an availability of pre-processed data from WP3 and WP4. WP3 provides documents, messages and news from relevant chosen data sources indexed with disease, symptom and location information. WP4 aggregates the documents into signals and creates new items to be recommended. WP5 uses that information together with the signal definitions of the users to produce recommendations to them.

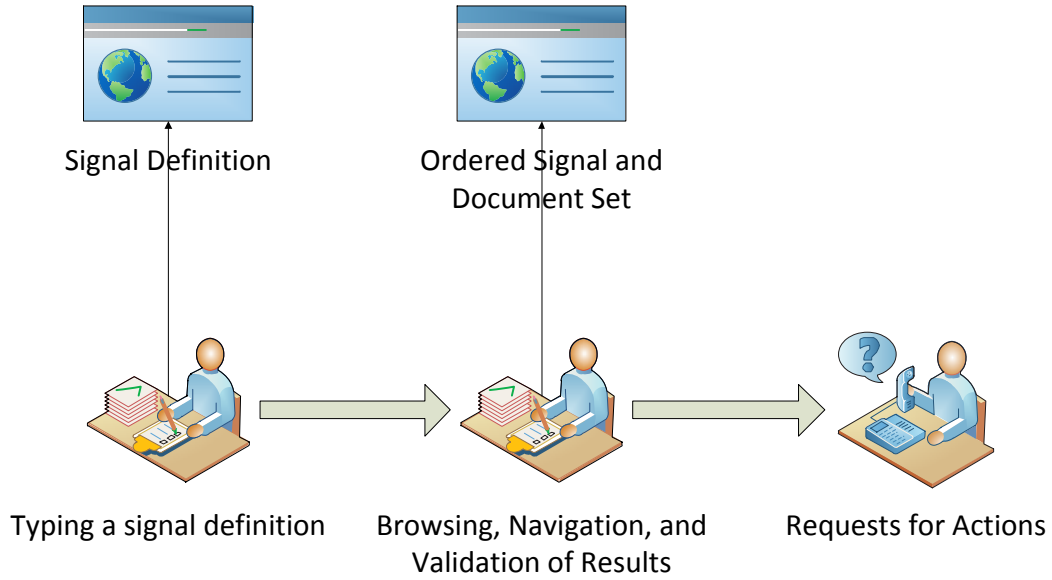


Figure 1: A medical surveillance user workflow concerning the signal definition, retrieval and navigation

Signal definitions are the main source used to build a user or a group profile. Such a profile is used to encode the user preferences and needs in forms of models which will be introduced later in this deliverable. It is then used to produce recommendations to the user. Besides the signal definition, WP5 also uses the user tagging activity and ratings as aspects to profile the user. These were already considered in methods described in the previous deliverable.

The main contributions of this deliverable are as follows. In addition to what we have already described in terms of the user profiling in previous deliverable, we extend the profiling with the *user's stated preferences for a set of locations*. We also propose *new approaches for recommendations to groups* and *navigation through the items in the system using tags*. The extensions and new methods address diverse aspects of the M-Eco system such as motivation scenarios, configuration management, formal models, implementations and some real use case scenarios on how personalization can be applied to support medical surveillance. The result of personalization can be seen, for example, as a reordering on the ranking of recommendations or search results or a personalized tag cloud leading users to quick access the information fulfilling the users interests.

These contributions will be covered in the remainder of this deliverable. It encompasses motivations, use cases, and scenarios describing benefits of personalization in M-Eco, design

and architecture of components, underlying models and their evaluation and deployment requirements. The content of this deliverable is organized as follows:

- Section 2 presents the motivations and assumptions for demonstrators, use cases and methods which utilize the personalization and adaptation techniques researched in this workpackage. It presents a concrete motivational medical scenario where users benefit from personalization.
- Section 3 presents design and architecture of personalization components including packaging. This section serves as an overview on what was delivered in the past deliverable, what is delivered now and what is planned to be delivered in forthcoming half a year.
- Section 4 introduces the personalization and adaptation models studied in this workpackage.
- Section 5 presents the evaluation of the personalization and adaptation components from user and performance viewpoints.
- Section 6 presents the requirements, installation guidelines and configuration for using M-Eco WP5 components and demonstrators either as a stand alone applications or through web services.
- Section 7 discusses the work delivered in the context of related work.
- Section 8 concludes the work, outlines the major achievements and points out future works.

2 Motivation

One of the main tasks of medical surveillance personnel is to identify whether there is a risk of an epidemic outbreak and, if this risk has high significance, act upon this. M-Eco works towards the goal of supporting this task by providing means to detect and suggest those signals and documents which reflect upon such emerging situations from (social) web sources.

This is done in M-Eco addressing three limitations of traditional systems:

- Recommendations are used to help users to deal with the huge amount of items in daily activities.
- Groups are used to help users interact with one another, share information, and discover new items that they would not have noticed otherwise, if working alone.
- A personalized navigation assists the users in efficiently browsing through the items and in finding related items.

These three limitations are addressed by personalization and adaptation methods in following demonstrators and **use cases**:

- Signal and document recommendations;
- Modeling group work interactions;
- Navigating through documents for validation of signals.

In the remainder of this section we first describe advantages and assumptions which directly resolve the limitations described above. Then we describe how the mentioned three use cases address the requirements from *M-Eco Deliverable D2.1* [3]. After, we describe a concrete motivational scenario for personalization and adaptation methods. Finally, we detail each of these use cases in the context of the scenario presented.

2.1 Advantages and Motivations for the Personalization and Adaptation Methods

Number of Users. Regardless of the number of users in the M-Eco System, recommendations are still necessary for each one of them. Both WP3 and WP4 methods, in spite of pre-processing, result in still huge amount of items to deal with. This happens because different organizations address information about diseases differently. The measure of severance or the granularity to tackle the available information vary from one institution to the other. Therefore, what is relevant for one might not be relevant for the other.

Besides the large quantity of items, the surveillance tasks are organized differently in each country and at the European and World level hierarchically. This means that a specific health official as a potential user of the system looks only at specific diseases affecting specific locations.

For this reason, WP5 deals with presentation and information overload issues by providing only those items (documents or signals), which are relevant for particular users according to their interests. This also means that, even though we do have even finite number of users provided by Robert Koch Institute (RKI) and the Governmental Institute of Public Health of Lower Saxony (NLGA) institutions in the project, they still require recommendations in order to be able to assess the relevant items for them. Even though these users know very well the information they deal with, recommendations will ease the process of selecting what is relevant and should be prioritized.

For example, to demonstrate the functionality of methods developed in WP5, in our evaluations (see Section 5) we show that users do not get overwhelmed by receiving documents related to some identified cases of outbreaks. Instead they receive recommendations that are relevant for the role or task in such organizations.

On the other hand, the tools and methods which we are developing have larger impact, as the methods work also for the cases with larger number of users which is especially relevant for cases of World Health Organization (WHO) and European Centre for Disease Prevention and Control (ECDC) are on the project advisory board. This include, for example, the methods which deal with user groups, helping individuals work in a collaborative fashion.

Collaborations between Users. Individual users can benefit from working in a collaborative environment, sharing information in groups. The methods which we are developing also support group work. By group work we mean the situations when several colleagues work together on the same or similar surveillance tasks to share the workload and find insight from each other activities. By doing so, we introduce groups in a twofold manner: i) one *explicit* formed by roles in organization and ii) the other *implicit* based on the user's interaction on recommendations.

The first group type refers to a role in an organization which can be executed by number of persons. In this case, the user profile and its signal definitions are shared and defined collectively and the generated signals and documents in a recommendation set are also shared. The support for such situations is then provided through commonly shared login for users in a specific role.

The second group type refers to a group environment where different users can interact with the recommended items. This is something derived from social media where users “follow” the group environment and interact with its items. Based on these interactions, the system can then assess the topics that are likely to be of interest to the group (and not only one individual) and recommend items that would otherwise be hidden from an individual user preference. The preferences and interests of those users are then calculated as an overlap under different chosen strategies in addition to the defined signal definitions.

A Need for Navigation Means. Users can find the information they are looking for faster and discover new items with a personalized navigation scheme.

Despite good precision and recall, there is always a need for means to support human intervention, understanding what system generates, and validation of signals and documents. This is typically well accepted when there is a sort of browsing support through the generated documents and signals. Both, documents and signals, are indexed either in WP3, WP4, and WP5 with additional metadata represented in the form of tags which can be organized in so called tag clouds.

Such navigation means also allow for reducing information overload and provide a user with a possibility to click only specific sub parts of document and signal sets to be examined and accessed through click on simple word labels representing diseases, symptoms, locations, but also on additional labels coming from the content of the documents. In the future, labels will also come from other users. This allows the user to navigate through the documents and signals using a set of keywords (i.e. tags) that can also help him or her discover new content, because these keywords were not necessarily written by the user but extracted from the documents.

2.2 User Requirements Addressed

The work being performed and delivered by WP 5 meets and satisfies the following requirements from WP2 and deliverable D2.1. Table 1 addresses the non functional requirements and how they have been fulfilled so far. Table 2 addresses functional requirements and how they have been fulfilled so far by WP5.

2.3 Personalization Scenario

To demonstrate our component and test our methods, we utilize the same scenario used by other work packages, based on the Enterohemorrhagic *Escherichia coli* (EHEC) outbreak in Germany. The first cases began to be reported in May and involved cases of Haemolytic Uremic Syndrome (HUS) and bloody diarrhea associated with infections by EHEC of serotype O104:H4.

The RKI was informed officially about three pediatric HUS cases in the city of Hamburg on May 19th 2011 [32, 19]. After follow-up reports stated that the number of cases continued to rise, an outbreak investigation involving all levels of public-health and food-safety authorities was initiated in order to identify the causes of the disease and to prevent further cases. Only in Germany, between 1st of May and 28th of June 2011, 3,602 cases attributed to the outbreak were reported to RKI. By end of June, 47 persons had died [19]. In the end, this turned out to be one of the largest described outbreaks of HUS/EHEC worldwide and the largest in Germany [18].

In the M-Eco system, this outbreak was used first by other work packages in order to assess its capabilities in detecting relevant information from social media and in triggering alarms

Table 1: Addressing non functional requirements of MEco by WP5

Requirement from D2.1	Description of Fulfillment	Level of Fulfillment
NFR 02 Information on Time: Information is accessible early enough so that it can be compared to information already gained from indicator-based systems.	In the <i>M-Eco portal use case</i> the signals and documents are recommended based on particular specified time interval. In the <i>twitter use case</i> the documents and signals are recommended for particular day, and they are always added in a time line order. In the <i>tag cloud use case</i> the documents and signals which are accessible by navigation in the tag cloud can be provided for specific time period.	Fully if time period is specified by a user.
NRF 03 Specificity of information: Information overload is avoided and noise is reduced. Amount of irrelevant signals is limited so as not to overwhelm user or to be too time-demanding.	In all three envisioned use cases the signals and documents are recommended based on location and disease. The other items are filtered out and not shown to the user. This reduces cognitive information load to only relevant documents. Evaluations show that this was achieved by relatively high precision. Further, the tag cloud allows for further reduction of presented items by clicking on particular tag in the tag cloud.	Fully with respect to signal definition.
NRF 04 Information collection and management by the system is personalized and self-learning: Information about a user and history of user action / interest is managed based on user profiles, and system can self-organize based on specific parameters.	In all three envisioned use cases the demonstrators observe user activities of various kind such as rating, tagging, clicking, commenting, and so on to infer interests additional to signal definition. These are used in personalization and adaptation algorithms. This allows the system to learn and adapt the recommended items to better suit user activities. The details can be found in the section on adaptation and personalization models.	Fully with respect to considered aspects of a user.

before other early warning systems. Documents were extracted by WP3 matching references to the diseases related to the outbreak and their corresponding symptoms. Figure 2 shows the number of documents on EHEC per day identified by WP3. Next, these documents were aggregated into signals with methods used by WP4. We then used the output from both work packages to test our methods in the context of this scenario.

In all three use cases, the same signal definition is considered. It consists of rules related to EHEC in Germany. Figure 3 depicts a screenshot for entering such a signal definition.

Table 2: Addressing functional requirements of M-Eco by WP5

FR 01 List of Signals: List of generated signals is presented. Associated, embedded documents is visible when a signal is expanded.	In the <i>M-Eco portal use case</i> this is achieved by integration with WP4. WP5 methods contribute with ranking and ordering of signals. This is achieved by presenting signals and documents together on a time line or the same item list in the <i>twitter and tag cloud use case</i> . This is due to the fact that twitter and tag cloud are build with different assumptions on the use and can be therefore seen as additional views on information which can be explored along different dimensions.	Fully in M-Eco portal. Partially in twitter and tag cloud.
FR 03 Date of document: A document is time-stamped based on its publication online.	In the <i>M-Eco portal use case</i> and the <i>tag cloud use case</i> the date of the document is displayed as an attribute. The document dates are on the twitter timeline in <i>twitter</i>	Fully.
FR 09 Commenting: User can comment signals and documents.	In the <i>twitter use case</i> users can comment by retweeting.	Fully in Twitter. For other use cases planned.
FR 12 Visualization using innovative techniques: Tag clouds for intuitive exploration of documents; Maps for geographic representation, Time series show timestamps of sources.	<i>Tag cloud use case</i> implements directly one visualization. Twitter helps in presenting the items as a time line.	Fully for tag cloud. Partially in twitter. Map implemented by another WP.
FR 13 Filtering, sorting finding: The user can sort or filter the documents by sources, time, place, disease or symptoms.	Automatic filtering and sorting is implemented by WP5 algorithms with respect to signal definition for all use cases. Tag cloud allows for further filtering according to additional tags.	Partially by WP5. Fully in collaboration with WP7.
FR 15 Collaboration: Signal definitions can be shared among several users to avoid definition of the same signal by multiple users.	The notion of groups introduced in WP5 models and algorithms, and supported by the <i>twitter use case</i> was especially introduced to satisfy this requirement. Followers functionality of twitter meets this fully.	Fully by twitter use case.

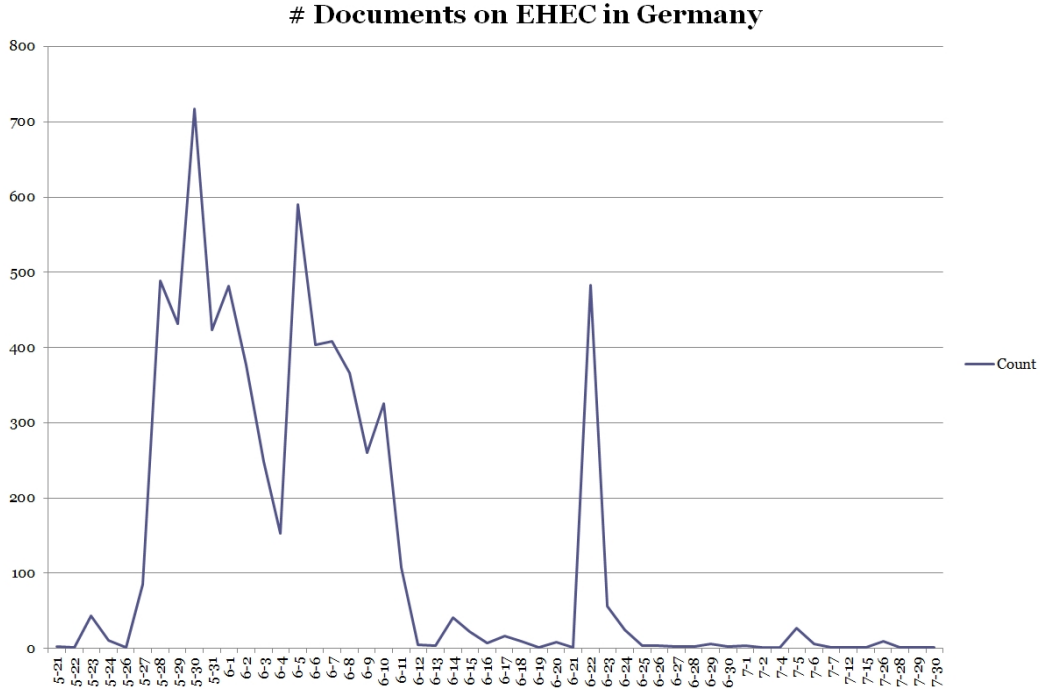


Figure 2: Number of documents identified by WP3 during different days of EHEC outbreak in Germany.

With the signal definition created, we assume that a M-Eco user, which works for a surveillance institution concerned with this outbreak, has the responsibility to monitor and validate possible EHEC-related events in different locations in Germany. The system, then, should be able to recommend to this user the relevant items detected previously by other work packages, promote collaboration among other users also focused on the outbreak, and help these users navigate through the signals and documents recommended to them.

These three goals are described next in the use cases. They make use of user models and recommendation methods described in later chapters. The user models take into account the user preferences based on their signal definitions (in this case, EHEC in Germany), tagging activity, and document ratings. These are then used by the recommendation methods to rank the items presented to the user and to suggest new items related to his or her modeled interests.

One key method described in this deliverable is the spatial reasoning, which can help users discover related outbreaks in other locations. For example, not only documents in the Bavarian region will be recommended to the user but also those that report on the surroundings or related areas of Germany and neighboring countries. This can provide the user with more relevant information besides what was predefined in the signal definition.

Signal Definition - Step 1 » Step 2: Settings »

<i>Diseases and Symptoms</i>	<i>Region</i>	<i>Source</i>
Disease Clostridium difficile Creutzfeldt-Jakob-Krankheit Denguefieber Diphtherie E.-coli-Enteritis	Country Chile China Costa Rica Dänemark Deutschland	<input type="checkbox"/> Select all MediSys TV Radio Blog Tweets
Symptom Durchfall, nicht blutig Durchfall, nicht näher bezeichnet Erbrechen Fieber krampfartige Bauchschmerzen	Region Saarland Sachsen Sachsen-Anhalt Schleswig-Holstein Thüringen	

Figure 3: A screenshot for entering a signal definition.

2.4 Signal and document recommendation through M-Eco portal

Signal and documents of interest of the user are recommended in the M-Eco portal. The personalized recommendations are triggered by analyzing signal definitions made by the user and his tagging activities. Signals and documents which aligned with to the user's interests are recommended (see Section 4.4).

The signal recommendation presents outbreak indicators including medical conditions, location where it was detected, the period in which the medical conditions were reported and tags assigned by users. The medical conditions are linked to the documents associated with signal. The document recommendations are composed by highlights of the document content, its source and a link to its entire content. By clicking on the link, the user is taken to another window on the browser to read the document. Figure 4 depicts signal and document recommendations in the M-Eco portal.

2.5 Modeling group work interactions

In this use case we consider the second group type presented earlier where different users can interact with the recommended items in a shared environment. Behind this type of collaborative or group work, there is the idea that the joint effort of many users sharing similar interests or goals has a higher potential to produce better outcomes [34]. In this context, recommendations of items can assist the group in their daily tasks and help them discover new items that would otherwise be hidden from individual user preferences.

Such recommendations can be useful when a group of individuals working together require relevant and up-to-date news information to support their decisions. In this case, teams need to assess new information in a timely manner in order to reduce undesired consequences of the ongoing event. A system making recommendations to these groups needs to: 1) *infer the group*

Signals		
Disease	Location	Date range
ehec, kopfschmerzen, durchfall	Germany	2011/05/01-2011/05/19
hus, darminfektion	Germany	2011/05/02-2011/05/12
ehec, kopfschmerzen, durchfall	Germany	2011/05/16-2011/05/22

Documents	
Source: 'aktuell.meinestadt.de'	Open in new window:
... [2011.05.30] Das Amt für Gesundheit in Frankfurt hat für Montag, 30. Mai, die Schließung der beiden Betriebskantinen eines großen Frankfurter Unternehmens aufgehoben. Alle Ergebnisse der Stuhluntersuchungen der Küchenmitarbeitern waren negativ. ...	
Source: 'Berliner Morgenpost'	Open in new window:
Neuer EHEC-Verdachtsfall in Berlin aufgetreten ... [2011.05.24] In Berlin wird ein fünfter Patient auf das gefährliche Darmbakterium EHEC und die Folgeerkrankung HUS untersucht. Eine Frau hatte sich offenbar in Norddeutschland infiziert. ...	
Source: 'dradio.de'	Open in new window:
Forscher arbeiten an neuem Therapieansatz für EHEC ... [2011.06.03] Der EHEC-Bakterienstamm, der in Deutschland bereits Menschen das Leben gekostet hat, stellt die Wissenschaft weiter vor ein Rätsel. ...	

Figure 4: A screenshot of signal recommendations (on the left) and document recommendation (on the right).

preferences from the actions of their members in the system, and 2) adapt to their evolving preferences.

However, one of the challenges of making recommendations to groups of individuals is to maximize the overall group satisfaction and, at the same time, keep individual dissatisfactions low [6]. To tackle this challenge, we proposed a method that generates group recommendations to a group of M-Eco users. The method infers the preferences of the individual users from their interactions with previously recommended items and builds a group model used to produce new recommendations. These take into account preferences in terms of the textual content of the recommended items (news articles), their locations and their time.

To test the method, we used the Twitter platform. Twitter is a microblogging service where users can create an account to publish short messages up to 140 characters. Users following that account have access to all messages published, without restrictions to a particular individual, and interact with them. The followers of an account, in essence, form a group of users interested in what is being published there. In Section 4 the reasons for choosing Twitter as one of the presentation options is discussed.

We use this notion of groups to simulate a group environment on Twitter where its members need to monitor the EHEC outbreak in Germany. In the evaluation described in Section 5.5, this outbreak is simulated with the actual data (as shown in Figure 2) identified by M-Eco, on their actual dates. Users are asked to interact with the published items in order for the system to determine the group preferences and recommend new items. Figure 5 shows a picture of the M-Eco account created on Twitter with the last 3 messages published.

Each message published contains two links. The first allows the members of the group (i.e. the followers of the account) to see the actual contents of the item published. The second link leads them to a tag cloud where they can see related documents to the one published on Twitter. This can help them discover new items related to the outbreak and browse more



Figure 5: M-Eco account on Twitter used to evaluate group recommendations.

efficiently through potentially large number of documents. Section 2.6 presents such scenario of navigation using tag clouds.

2.6 Navigating through documents for validation of signals

When a medical expert is assessing a particular signal to take further actions, he has to explore exhaustively all documents of that signal resulting into a time overhead. In order to illustrate this scenario, we show the case in which a considerable amount of document was detected by WP3 during the *E. coli* outbreak. At the peak of the epidemic, there were collected more than 700 documents per day (see Figure 2). Validating such amount of documents respectively is not feasible and there is a clear need to facilitate the assessing process of signals. Another issue of a signal validation is that a basic search given by the defined symptoms is not sufficient in cases when the medical expert is not aware of the possible keywords that can retrieve important documents. For example, during the *E. coli* outbreak in Germany as one of the possible sources of the epidemic was considered cucumbers from Spain. However, surveillance personnel does not have to be aware of these facts.

In order to address the above-mentioned problems, the M-Eco user has the possibility of navigation through tag cloud composed by tags that were assigned to the documents about *E. coli* with identified location Berlin. Such personalized tag cloud depicted in the Figure 6 provides a general picture of considered documents which enhances a validation of the signal by the expert. The depicted tags can be considered as additional symptoms of the given signal and in a such way it can significantly influence expert's decision or action. The personalized tag cloud is composed by tags "durchfall", "fever", "magenschmerzen", "darminfektion" which can be considered as additional medical conditions or symptoms of the given outbreak. The tag cloud contains also terms as "gurken", "tomaten", "bauern" which give a new insight into a given signal as it strongly indicates that the spreading of the infection can be caused by the



Figure 6: Tag cloud for E.coli outbreak in Berlin

consumption of particular vegetables. Another group of tags are related to the geographical names like “Berlin”, “Deutschland” or “Spainen” which can help to distinguish between specific information related to Berlin and documents that describe E.coli outbreak on country level or about countries that are involved into this outbreak for some other reasons e.g., Spain because of cucumbers.

Furthermore, the tag cloud can support serendipitous discoveries, i.e. tags in the cloud could be indexes to unexpected but relevant documents that help medical experts to assess a signal [55]. This particular feature provides new perspectives of assessing a given outbreak and therefore enhancing the signal validation process.

3 Design and Architecture

This section presents the architecture of WP5 framework including the component integration and packaging. The component integration section shows how components integrate with the external system and how they connect between themselves. The component packaging provides a logical view of packages that implement the components. This documentation overviews the personalization framework and facilitates its extension and reuse.

3.1 Component Integration

Figure 7 overviews the complete WP5 infrastructure divided in three layers: presentation layer, business logic layer and data access layer:

- **Presentation Layer** - is shown on the left part of Figure 7 and is represented by personalization functionalities (recommendations and tag cloud) implemented at the user interface of the following information systems: M-Eco portal ¹, Medisys Health Informa-

¹<http://139.191.1.59/m-eco/>

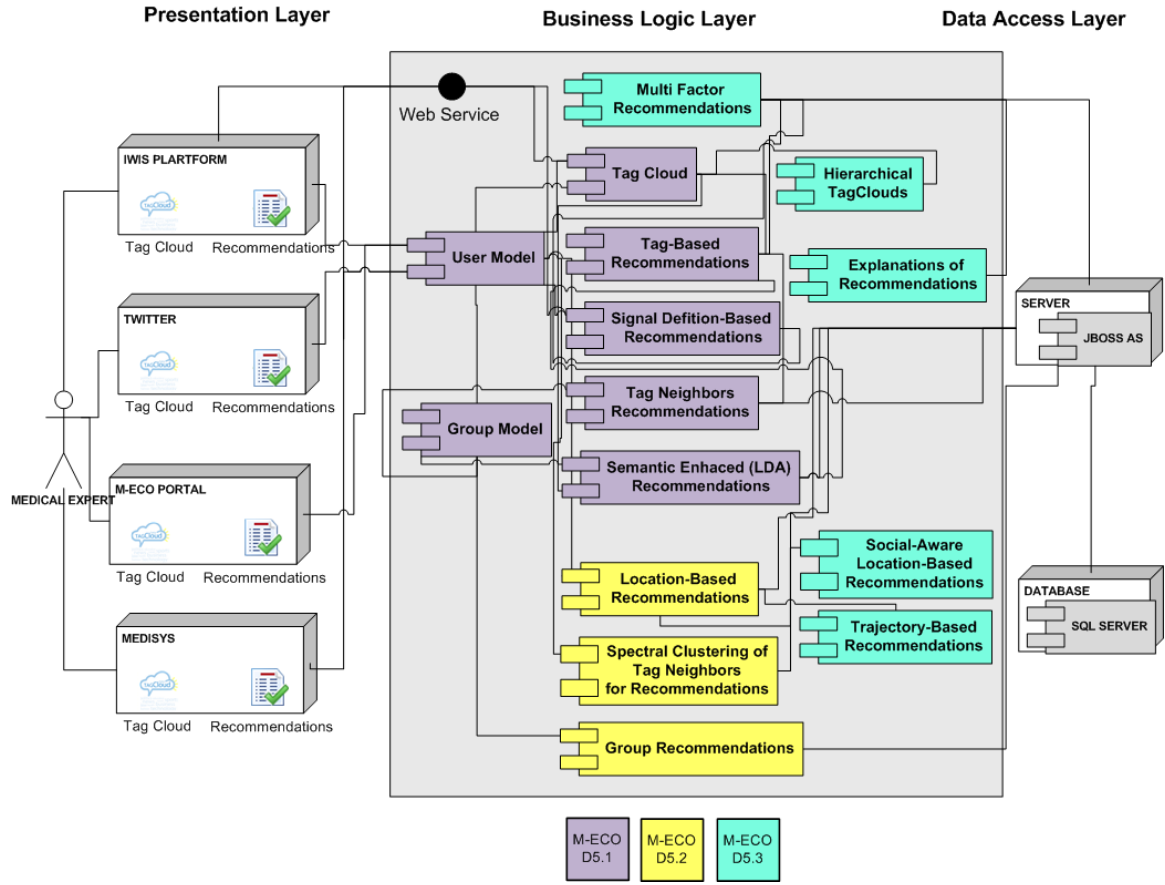


Figure 7: WP5 Component Integration.

tion System ², IWIS platform ³ and Twitter ⁴). An illustration of recommendations in the M-Eco portal is shown at the motivation section of this deliverable in Figure 4 and the illustration of a tag cloud in the IWIS platform is shown in Figure 6. Worth mentioning that the services provided by the recommendations and personalized tag cloud are also available through REST web services (details in Section 6.2).

- **Business Logic Layer** - is shown in the middle part of the Figure 7 and is represented by components that generate recommendations and personalized tag clouds to the information systems at the presentation layer. It also shows how the components interact with each other. At first, the user and group model components build the user and group model with information gathered from the information systems in the presentation layer, then these user models are utilized by the recommendation and tag components for pro-

²<http://medusa.jrc.it/>

³<http://demos.iwis.cs.aau.dk:8081/WP5/home.seam>

⁴<http://twitter.com/>

ducing the personalized recommendations or tag clouds. The components are rendered in three different colors distinguished by the M-Eco deliverable where they are described. The components colored in *purple* were described in the M-Eco Deliverable D5.1 [60], the components colored in *yellow* are in implemented for the current M-Eco Deliverable D5.2 and the components colored in *turquoise* are planed to be implemented in the M-Eco Deliverable D5.3.

- **Data Access Layer** - is shown on the right part of the Figure 7 and is represented by components that maintain and access data on the database. The components in the business logic layer manage transactions to the database including storage, updates and removals. The components in the middle layer connect to the Microsoft SQL Server database through the JBoss Application Server. Worth mentioning that we are changing the current database software to PostgreSQL 8.3 due to license constraints.

3.2 Component Packaging

The component package presents a logical view of packages (and their connections) that contain the actual model implementation of the components shown in Figure 7.

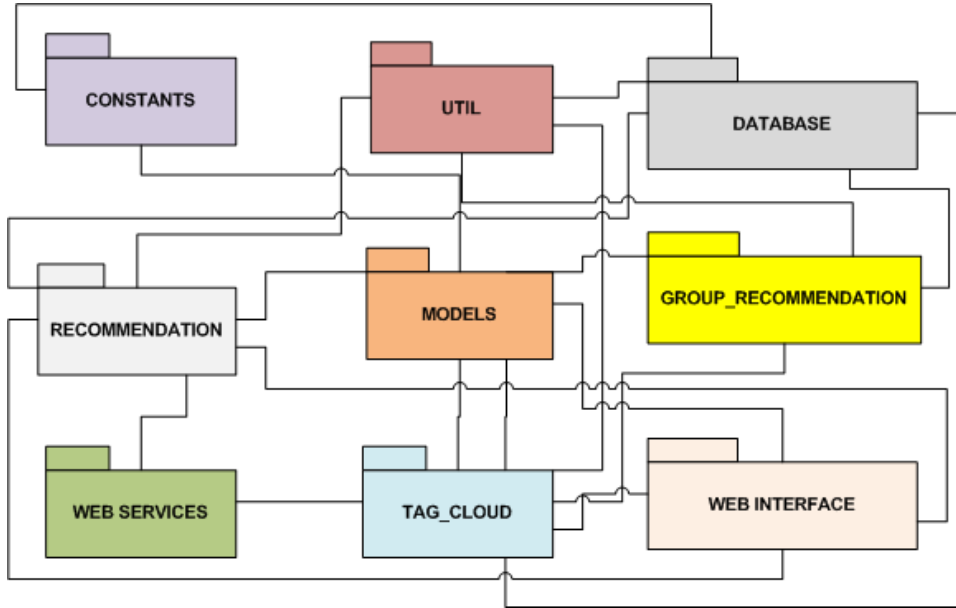


Figure 8: WP5 Component Packaging.

- **Model** - Package where the conceptual classes are placed such as Document, Indicator, User, Signal, etc. The individual and group models are explained in Section 4.2.
- **Recommendation** - Package where the classes that implement the personalized recommendations. This package utilizes classes from package Model and Database. The

recommendation models implemented are *Signal Definition-Based Recommendation* (Section 2.2 of M-Eco Deliverable D5.1 [60]), *Semantic Enhanced Recommendations* (Section 2.4 of M-Eco Deliverable D5.1 [60]), *Tag Neighbors-based Recommendations* (Section 2.3 of M-Eco Deliverable D5.1 [60]), *Spectral Clustering of Tag Neighbors for Recommendation* (Section 4.3.3), *Tag-Based Recommendations* (Section 4.4.2) and *Location-Based Recommendation* (Section 4.4.3). We also include here the future implementations such as *Social Aware Location-Based Recommendations*, *Multi-Factor Recommendations* and *Explanation of Recommendations*, all to be described in M-Eco Deliverable D5.3.

- **Group Recommendation** - Package where the classes that implement the recommendations addressed to groups. This package utilizes classes from package Model and Database. The group recommendation model implemented is the *Group Recommendations* (Section 4.4.4).
- **Tag Cloud** - Package where the classes that implement the personalized tag cloud. This package utilizes classes from package Model and Database. The personalized tag cloud model is explained in (Section 4.4.5). We also include here the future implementation *Hierarchical Tag Cloud* to be described in M-Eco Deliverable D5.3.
- **Web Service** - Package where the classes that implement the RESTFUL web services providers and the web service clients (Section 6.2).
- **Util** - Package where the utilitarian classes are for string and file processing. Classes from all packages, except the Model one, access classes from the util package.
- **Constants** - Package where the widely constants are placed such as database url and credentials.
- **Database** - Package where the classes that handle the database access as well as queries are stored. Classes from all packages, except the Model, Util and Constants, access classes from the Database package.
- **Web Interface** - Package where the classes that handle all assets related to the user interface. This package communicates with recommendation, group recommendation and tag cloud packages.

4 Adaptive Tuning and Personalization Models

This section presents the personalization methods proposed for solving the problems introduced in the Section 2. In order to address the issue of *users dealing with the huge amount of items in their daily activities*, we provide an analysis of presentation options so that the user interface is made up with intuitive and friendly components. In addition, we propose a number of recommendation methods for selecting the most appropriate piece of information by filtering out noisy and unimportant content. In order to support *groups of individuals that share similar interest with relevant information*, we provide means of combining individual preferences into group profiles so that they receive recommendation of common interest. In order to help users *navigating through vast collection of documents and finding new items*, we provide a visual representation of documents through a tag cloud component. Besides indexing documents in the corpus, each tag helps users to find new related information of interest. The following

subsections present different models used to address each of the above problems with a particular contribution. These methods are also evaluated in user studies or controlled environments as shown in Section 5.

4.1 Analysis of Presentation Options for Recommendation and Adaptation

The analysis of presentation options aims at investigating techniques for adapting, structuring and browsing information in an intuitive and friendly way to the end user. For the M-Eco prototype this analysis is important because it provides means of easing the work of medical experts while exploiting or navigating through documents in the system. In order to support medical experts in this exploitation process, we outline *three* important aspects that user interfaces should consider:

- The interface should browse relevant information and reduce information overload for users. For instance, a subset of relevant items should be displayed rather than a vast collection of possibly important documents.
- The interface should present a timeline of signals and documents in order to provide a time overview for the displayed resources and in a way that facilitate user orientation and exploitation of a vast number of signals or documents.
- The presentation layer should facilitate the exploitation of a large number of documents with alternative information views to provide an overview of a given set of documents. For instance, simple listing of related documents is sufficient when the number of such resources is small. When there is a need to exploit a larger set of documents the interface should provide a navigational tool that present a general picture and important properties of these documents. Such navigational component should lead a user only to important and relevant subsets of the documents and minimize an overhead from exploring all documents.

The above considerations will impact directly on how the M-Eco prototype will design its user interface. Therefore, M-Eco interface provides a relevant information for a particular user according to the signal definition which expresses user's interest. The interface presents all signals that match medical conditions and location data from the signal definition. In this way, an information overload is significantly decreased. However, when no or just a few matching signals exist there is a need to present additional similar signals to the defined user's preferences. Such recommended signals provide an additional context for user's preference which facilitates to better evaluate and assess a considered outbreak. Signal suggestions that have similar medical conditions or location to those defined in user's signal definition are computed according to a multifactor recommendation model. Signals are sorted according to the computed ranking and only first top 10 (most similar) signals are presented.

In case, the M-Eco prototype would not present recommended signals, a given user would have to define additional signal definitions similar (in terms of medical conditions or location) to the initial one. Exploration of signals matching to additionally introduced signal definitions would lead to time overhead and confusion of the user. Hence, there is a clear need to present a list of recommended signals in the M-Eco interface. In case a whole list of suggested signals presented in M-Eco prototype was explored by the user additional recommended signals should be available.

A timeline of relevant documents and signals is another important feature, but currently not yet supported by the M-Eco prototype. To demonstrate such functionality and minimize a development effort we explored available interfaces. We have decided to utilize a social networking service Twitter which also provides a user feedback in different ways. The interface of Twitter provides a timeline of short messages (less than 140 characters) called tweets that were posted by users. Interesting tweets can be marked as favorite and also shared by other users of the social system. We utilize Twitter as the presentation tool that provides a list of recommended documents and signals ordered by time. The interface can be easily linked or integrated into M-Eco prototype. The recommended items are published by web services so the same functionality can be achieved with any other GUI. On the other hand, there are some limitations of that approach as Twitter is another independent system which is maintained by third-party company which can result in security risks or even sudden unavailability of the service.

Moreover, the M-Eco interface should facilitate an exploration of a large number of documents related to given signals. This browsing and retrieving process can be supported by the following retrieval interfaces:

- Graph-based query refinement tools - users can retrieve a subset of documents by exploring a graph where a node represents a concept and an edge indicates a relation between two concepts. The drawback of these methods is that concepts are given by the referenced terminology. To illustrate such approach, two different interfaces are depicted in the Figure 9. The Cat-a-Cone interface supports an exploration of large hierarchies of pre-defined terms with corresponding sets of documents (Figure 9a). TileBars is a visualization tool which represents retrieved documents with horizontal bars where rows represent an occurrence of the searched terms within the document (Figure 9b). This approach indicates relative length of documents, frequency (distribution) of search terms in the document but also with respect to all retrieved documents.
- Clustering - documents are grouped based on some similarity measure and each cluster represents a subset of documents that share some common properties. The labeling of obtained clusters is the main limitation of this approach. An example of visualising clustered documents can be seen in the Figure 9f where each cluster of documents is represented as circle. Cluster can contain nested subclusters. Second example illustrates a cluster analysis over web pages related to entertainment (Figure 9e). Each region in the map represents a specific labelled cluster with a certain number of assigned documents. The same document can belong to multiple clusters.
- Faceted search tools - documents are divided into pre-defined categories (each category represents a particular facet). It allows an exploration of specific subsets of documents. The problem is that facets have to be pre-defined in advance and it can limit the discovery of unexpected subsets of documents. Aduna Autofocus enterprise search system provides a standard textual faceted navigation interface where on the left side of the page are displayed different facets (Figure 9c). Another example of the faceted search interface is RelationBrowser depicted in the 9d.
- Tag cloud - a subset of documents is retrieved by a clicking on the term available in the interface. Terms called tags displayed in the tag cloud are defined by users or automatic annotation tools. Therefore, various combinations of tags result in different subsets of retrieved documents and all emerging trends and relationships in the available set of

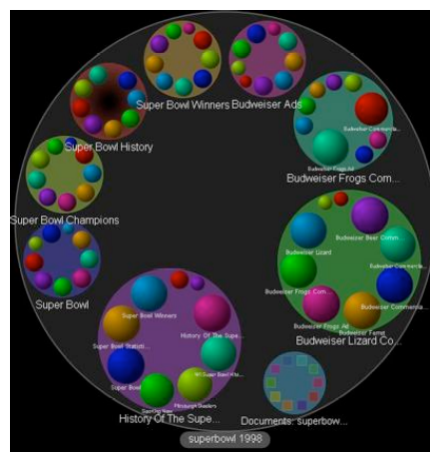
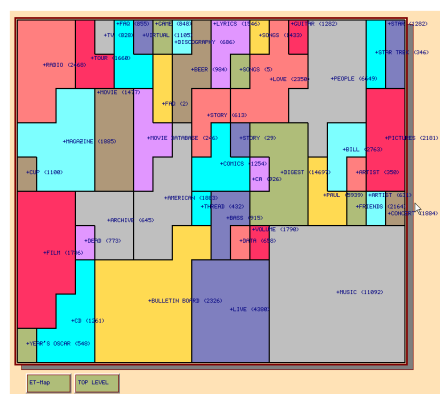
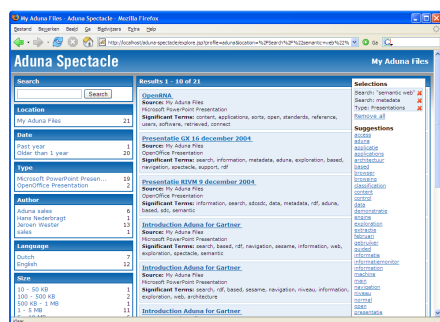
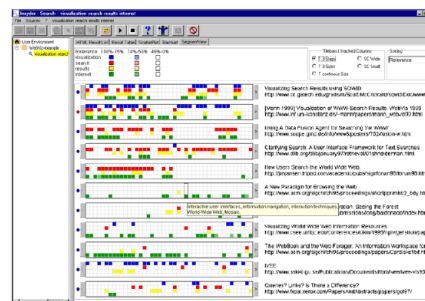
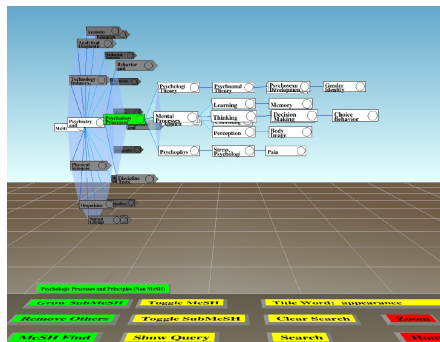


Figure 9: Visual comparison of different retrieval interfaces. Depicted pictures are taken from the book “Search User Interfaces” [25], Chapter 10.

documents can be explored. An example of this visual interface was presented in Section 2.6 in the Figure 6.

The above-mentioned presentation options are more analyzed in [26, 25]. The M-Eco prototype aggregates documents from different sources with different nature - therefore, a process of pre-defining concepts or facets is not feasible. Because of that, a tag cloud interface is chosen for summarization and query refinement tasks. A tag cloud supports a retrieval of specific subsets of documents so that the exploitation of vast amount of documents is improved.

Each relevant signal displayed in M-Eco prototype can be explored by accessing a pertaining tag cloud. This facilitates a validation process of a particular signal. Further, a tag cloud is integrated also with Twitter interface where each recommended document or signal can be further explored by visiting a related tag cloud. Integration of tag cloud service with M-Eco prototype and also with Twitter based recommendations provides an alternative and supplementary way of browsing and retrieving of specific documents. Worth mentioning that the tags selected for the cloud formation are subject of relevancy analysis so that the final tag cloud widely covers the set of documents in the corpus. We provide a user study about an importance and usability of tag cloud in Section 5.3.

4.1.1 User Feedback Options

In order to improve the performance of our personalization models, the personalization models need to collect the user's feedback constantly on the items recommended or browsed at the user interface. Thus, we analyzed a number of feedback options that were eventually applied either in our components or in the experimental evaluations of our components. Section 5.1.2 shows a group discussion where we evaluate the assessment of our recommendations.

- *Thumbs-up or thumbs-down* is a hand gesture with the thumb extended upward or downward in approval or disapproval, respectively. This provides a binary assessment of an item recommended with the meaning positive for thumbs-up and negative for thumbs-down. The disadvantage is that a user may be ambivalent and none of the options will actually represent his feeling.
- *Rating Scale* is a means of assessment in terms of quality, quantity, or some mix of both. For example, one to five stars is commonly employed to categorize hotels. In our experiments, we extensively utilized a 5-star rating scale meaning 1-(irrelevant), 2-(little relevant), 3-(average), 4-(relevant), 5-(very relevant).
- *Comments* is a means of users describe their impression about the system itself textually. The advantage of this method is that they are not constrained by a set of alternatives. The drawback is that only a few users voluntarily provide comments.
- *Comments* is a means of identifying the user's interest in a given topic or tag by account the amount of clicks an item sums up in a period of time. This is an implicit way of collecting users preferences so that they are not instructed to act.
- *Bookmarking* is a direct indicator of a user's preferences once the user explicitly determines what his preferences are. This premise is more reliable when the bookmarks are labeled as "Favorite" or related meaning.

4.2 User and User Group Model Definition

User Modeling refers to the activity of maintaining information about the user's interest, abilities, knowledge and goals [7]. Information systems adjust their content, data, business rules, user interface according to the user's model information [59]. The performance of personalization components depends on how well elaborated the user model is. Information systems that personalize information for individuals sharing common interest need to create group models instead of single user models. In this case, the group needs to suppress the individual needs. Our personalization component develops user models for individual users and groups.

In general, our user models combine multiple indicators of user preference. For example, a system can learn a user's preferences based on the most frequent criteria of his signal definitions but also through his tagging activity. In this case, these two factors can be combined to generate more accurate recommendations to this user. The complete explanation on how this method is implemented and performs on real world data set is explained in *Section 2.2 of M-Eco Deliverable D5.1* [60].

4.2.1 Tagging Model

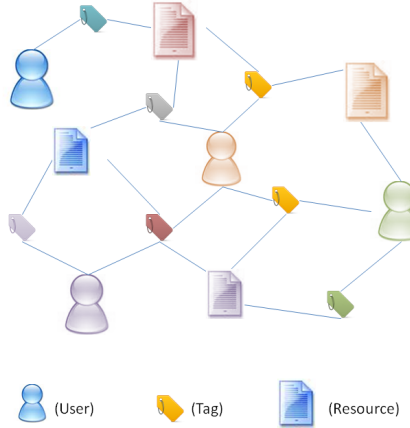


Figure 10: Graph that illustrates the social tagging activity.

Most of personalization in the M-Eco project relies on the user's tagging activity. We understand that tags are potential sources for learning user's interests and therefore can be utilized for selecting appropriate pieces of information respecting individual preferences. Meanwhile tags help solving the problem of users dealing with huge amount of items, tags can help users to navigate through documents of interest through personalized tag clouds. Before depicting the personalization models that address the problems listed above, it is necessary to understand how taggings are modeled and represent the user's interest.

Our general personalization model (based on [30]) realizes tagging systems as hyper graphs where the set of verticals is partitioned into sets: $U = \{u_1, \dots, u_k\}$, $R = \{r_1, \dots, r_m\}$, and $T = \{t_1, \dots, t_n\}$, where U, R, and T correspond to users, resources, and tags. Figure 10 illustrates the

social tagging activity. In the context of M-Eco, the resources can represent signals, indicator, documents or any other entity which can be assigned with tag.

A tag annotation, i.e. a resource tagged by a user, is an element of set Y , where: $Y \subseteq U \times R \times T$. The final hyper graph formed by a tagging system is defined as G with: $G = \langle V, E \rangle$ with vertices $V = U \cup R \cup T$, and edges $E = \{\{u, r, t\} \mid (u, r, t) \in Y\}$. Particularly to understand the interests of a single user, our models concentrate on the tags and resources that are associated with this particular user, i.e. in a *personal* part of the hyper graph G . We then define the set of interests of a user as $P = (T_u, R_u, Y_u)$, where Y_u is the set of tag annotations of the user: $Y_u = \{(t, r) \mid (u, t, r) \in Y\}$, T_u is the tag set of the user: $T_u = \{t \mid (t, r) \in Y_u\}$, R_u is the set of resources: $R_u = \{r \mid (t, r) \in Y_u\}$.

4.2.2 Group Modeling

In every day activities individuals usually work collaboratively where one benefit of the other's work. Users are constantly sharing information and working together on team oriented tasks to achieve common goals. Because of collaborative work and other personal reasons, it is natural that individuals share common interest and multiple information can be addressed to groups instead of single individuals. Given this premise, it is reasonable that our personalization models support the team work and do not constraint themselves to individual recommendations. We find necessary to understand how groups are modeled so that personalization is effective to them.

In particular, we investigate how to recommend information of interest of groups of individuals following a microblog account on the Web. The group model is created from a combination of individual user models that can be used to guide item selection and recommendation. In the following, we show how individual models are combined into a group model. We represent the user profile by the tuple $\{message, f, w\}$, where:

- $message$ represents the message published in the microblog account followed by the user or published by himself in his own account;
- f is set to 1 if the message is marked as favorite by the user, or 0 otherwise. Messages marked as favorite are assumed to be an explicit indicator of the user's preference;
- w is set to 1 if the message is forwarded to the user's own microblog account, or 0 otherwise. Although not a strong indicator such as f , we also consider a forwarded message as an indicator of user preference since this message is shared with the user's group of friends/followers.

Provided such a user profile, we define the *group profile* in our approach as the vector $GP = [message, value]$ of messages of interest to the group and the set IM of messages ignored by the group, i.e. where all members of the group did not interact with the message. In GP , $value$ represents the utility value of the message for the group, following the utilitarian strategy, i.e. utilizing the *average* of the preferences of all the group members. The $value$ is calculated from the individual preferences as:

$$value(message) = \sum_{i=1}^n (f + w), \quad (1)$$

where n is the total number of users following the microblog account that interacted with the *message*. In the end, GP will contain all messages published in the microblog account that at least one user interacted with. We also take into account the messages published in the group recommender account that no user interacted with, i.e. the ignored messages. These are assumed to be a valuable insight of group *disinterest*, although not necessarily dislike. All the published messages ignored by the group are added to the set IM .

Tag-based Group Model. An alternative way of profiling groups of individuals in social tagging systems is shown in *Section 2.5 of M-Eco Deliverable D5.1* [60]. In this approach, user and document profiles are grouped into several groups characterized by the significant tags, with the belief that common tags annotated by the most objects inside the group can reflect the characteristics of user preference or document functionality.

4.2.3 Location Preference Model

In order to recommend users relevant information it is necessary to assess whether a document matches the user’s interest. However, a number of related (and relevant) information still can be left behind. For instance, many documents matching a signal definition with the specified location *Munich* can be quickly retrieved while relevant information reported by neighbor locations (e.g., Augsburg, Rosenheim and Salzburg) are not considered. In order to improve the recall with related documents, we investigate spatial reasoning techniques as a means to help users discover related outbreaks in other locations, related to user’s stated references.

In order to suggest items of related locations, we first model the user’s location preferences. These preferences are initially modeled from the locations specified in the signal definition SD_u of the user u . We define the set L_u , the initial set of locations the user has an explicit preference as:

$$L_u = \{l | l \in SD_u\}, \quad (2)$$

where l is any location the user can choose from in the M-Eco system. Next, we introduce the concept of location similarity in order to find related locations. This similarity is computed as a function of two factors:

- Political hierarchy (e.g., city, state or country level); and,
- Distance and population.

Once the similarity score between two locations is computed, we expand the set L_u with the most similar locations. For each location $l \in L_u$, we expand the set L_u with the locations with similarity score higher than 0.9, where the maximum score is 1 when the location being compared is the same as l . In the end, the final set L_u contains the user locations together with their most similar ones. This information is then used to compute recommendations of items from related locations to the user. Section 4.4.3 details how the location similarity is computed and recommendations are produced.

4.3 User Classification and Modeling Algorithms

In this section we present a number of techniques for modeling user generated data such as tagging in order to generate proper recommendations. The recommendations are intended

to reduce information overload of medical experts and facilitate their tasks while evaluating medical documents. In the following subsection, besides introducing the modeling algorithms, we highlight technical problem addressed and how they were solved.

4.3.1 Semantic Enhanced Tag-Based Recommendation

In order to lessening the problem of users when dealing with huge amount of information available, this approach aims at supporting users by selecting and retrieving relevant information that are particularly annotated with tags. For that, this method generates personalized recommendations from the analysis of the user’s tagging activity. In order to recommend users with personalized information, the approach deals with common problems in social tagging systems: sparsity (when no or few tagging are available) and ambiguity (when a single tag refers to different concepts). The recommendations model tries to capture the semantic nature of social tagging data and then incorporates the semantic-enriched expression along with the traditional lexicon-based vector for improved recommendations. In the end, we combine a conventional tag-based recommender systems with latent semantic analysis to generate personalized recommendations. The complete details of this model are demonstrated in *Section 2.4 of M-Eco Deliverable D5.1* [60]. The following subsection provides another method to support users with the selection and retrieval of pertinent information.

4.3.2 Expanding Tags with Neighbors For Improving Search Retrieval

This approach also addresses the problem of users dealing with huge amount of items. This method supports users with selection and retrieval of relevant documents so that they are not distracted with unsolicited content. From a different perspective, this method aims at assisting proactive users searching for items of interest instead of receiving recommendations in periods of time. Although search and recommendations are distinct mechanisms, the model for relevance assessment and retrieval is relatively common for both approaches. While recommendations look at the contextual information for suggesting information, the search engines depend on a query issue by a user. The technical problem addressed in this approach concerns under-specified queries that lead users to irrelevant search results. In order to tackle such an inconvenience, we proposed a method that utilize tags to augment the chances of relevant retrieval. In brief, when a user issues a query, it is extended with pre-computed related terms (called tag neighbors) aim at improving the retrieval of relevant items.

In order to test our approach, we crawled the articles from *MedWorm* repository system. The focus of our analysis was based on the observation of precision of our search engine. We compared our precision results with results from a baseline query search that rely on the simple user entry query (without expansion). The overall result was satisfactory, our results verify that our approach outperforms the baseline query search in terms of mean average precision (MAP). As a limitation of the work, we realized that the tag extension does not perform equally for all medical categories. As a future work, we aim at improving the quality of tag neighbors by comparing them against medical specialized dictionaries or domain ontology vocabularies. Further, we plan to realize more experimental studies necessary to validate the scalability and feasibility of the proposed approach in a broader scope. Finally, we aim at combining the current approach with other techniques previously explored such as collaborative filtering. The complete demonstration and evaluation on how this method can be seen in *Section 2.3 of M-Eco Deliverable D5.1* [60]. The following subsection provides an alternative method that also utilize tag neighbors for improving recommendations.

Algorithm 1: Spectral Tag Clustering

Input: The tag-user matrix and tag-document matrix, i.e. $TU = \{TU_i, i = 1, \dots, K\}$,
 $TD = \{TD_i, i = 1, \dots, K\}$

Output: A set of C tag clusters $TC = \{TC_c, c = 1, \dots, C\}$ such that the cut of
 C -partitioning of the bipartite graph is minimized

- 1 Construct the integrated tag-user-document matrix by stacking up the above two matrix, $T_i = TU_i \cup TD_i$;
 - 2 Calculate the diagonal matrices D of T ;
 - 3 Form a new matrix $RT = D^{-1/2}TD^{-1/2}$;
 - 4 Perform SVD (Singular value decomposition) operation on RT , and obtain k singular vectors L_s to create a new projection matrix RV ;
 - 5 Execute a clustering algorithm on RV and return clusters of tags in the integrated vectors: $TC = \{TC_c, c = 1, \dots, C\}$.
-

4.3.3 Spectral Clustering of Tag Neighbors for Recommendation

In line with previous method (Section 4.3.2), the primary goal of this approach is to support users with relevant recommendations. In order to improve recommendations with quality and quantity of tag neighbors, we investigate the spectral clustering algorithm to filter out noisy tag neighbors. The basic idea of proposed approach is to utilize the tag neighbors to extend the users' or documents' profiles which are represented by the tags. In the tag similarity matrix, each tag has different similarity weights with other tags, we assume the higher the weight, the more similar the tags are to the target tag. To realize the task of expanding the tag, the major difficulty is how to define the tag neighbors and how to locate them from the total tags. Here we adapt a statistical definition of tag neighbor - the tags which are co-occurred most frequently or they have the higher similarity weight in the similarity matrices to the target tag, are the neighbors for each other. So the N tags according to the top- N similarity weight can be defined as the tag neighbors of an individual tag. After such steps, each tag will have an additional neighboring tag set which will help to improve the quantity of the tag expression.

Given an arbitrary tag T , its neighboring tags are defined as:

$$Nb(T_i) = \{T_j : T_j \in TopN\{SM(T_i, T_j)\}\} \quad (3)$$

where $TopN\{SM(T_i, T_j)\}$ is the tags which possess the top- N highest similarity values to tag T_i .

Tag Neighbor Filtering based on Spectral Clustering. The tag clustering is used to find the tag aggregates with similar functions or topics. The spectral clustering is based on the graph partition which maps the original co-occurrence observations onto a new spectrum space with a reduced dimensionality. The obtained partition guarantees the disjoint clusters with minimum cut optimization. Especially in the context of social tagging systems, we know each tag can be expressed a column vector of user profile and document profile, i.e. $T_i = TU_i$ and $T_i = TD_i$. Then we stack up TU_i and TD_i and form a new tag vector over users and documents, $T_i = TU_i \cup TD_i$. At last, we employ spectral clustering on the integrated tag vectors to get tag clusters. The pseudo codes of the spectral tag clustering are listed in Algorithm 1.

Algorithm 2: Tag Neighbor Expansion for Recommendation

Input: A collected tagging data

Output: A list of top-N documents for the candidate user

- 1 Pre-process the tagging data to construct a user profile matrix and a document profile matrix.
 - 2 Represent the tag in the user profile vectors and document profile vectors.
 - 3 Calculate the tag similarity matrix.
 - 4 Get the top-N tags according to the highest values in the similarity matrix for each tag.
 - 5 Partition tags into different clusters by spectral clustering algorithm.
 - 6 Check all the tag neighbors generated in the previous steps whether they belong to the same cluster with the original tag, and filter out noisy tag neighbors.
 - 7 Update the tag vectors of user profiles and document profiles with tag neighbors.
 - 8 Calculate the similarity between the candidate user's tag vector and the document tag vector, and rank the documents according to the similarity values in a descending order.
 - 9 Select the top-N documents with the first N highest similarities as the recommendations to the candidate user.
-

From the above tag clustering, we obtain a set of tag clusters:

$$TC = \{TC_1, \dots, TC_C\} = \{(T_{11}, \dots, T_{1i_1}), \dots, (T_{C1}, \dots, T_{Ci_C})\}$$

The naive tag neighbors will be calculated by collaborative approach. However, the problem is that not all of the expanded tag neighbors are appropriate for the target tag since there are some noisy tags in the tag neighbors. From the above way, all of the tags are clustered into the several individual clusters where each cluster is a set of tags sharing the similar functions or topics. Bearing this phenomenon in mind, we thereafter propose to utilize tag clusters to determine whether the naive tag neighbors are included in the expansion of tags.

The basic idea of clustering filtering is: the neighboring tags from the same tag cluster of the target tag contribute collaboratively to specific function or topic, being kept as the appropriate tag neighbors for tag expression expansion; otherwise be discarded. So the next processing step is to filter out the noisy tags according to the discovered tag clusters. Worth mentioning that each tag has an expanded tag neighborhood, which might belong to different clusters. To ensure all neighboring tags are from the same tag cluster, each tag in the expanded neighborhood will be compared with all the tags from the tag cluster where the target tag is assigned. If the expanded neighbor appears in the same cluster, it then can be considered as the appropriate neighbor of the tag tag, making it kept in the expanded tag set; otherwise, it should be filtered out. After such steps above, the left elements could be defined as the tag neighbors for the target tag, and the quality of the tag neighborhood will be accordingly improved. Also in such way the density in the integrated tag-user-document matrix could be increased substantially.

For example, given the Tag T_i belongs to a certain cluster C_j , and the T_i has an expanded tag neighborhood as $TN = \{TN_1, \dots, TN_k, \dots, TN_K\}$, then we can compare each tag TN_k with the tags in C_j , if TN_k is existed, it can be defined as the tag neighbor; otherwise it will be discarded. After such steps, the tag neighborhood will be updated.

Improved Recommendation with Tag Neighbor Expansion. After the tag neighbor expansion is completed, we get updated user profiles and document profiles in the forms of

tag vector expression with expanded tag neighbors. We then utilize the similarity measure between users and documents to make tag-based recommendations. The whole algorithm for improved recommendation with tag neighbor expansion is described as Algorithm 2. A complete description of this work is seen in our research article [50].

4.4 Recommendation, Adaptation and Personalization Strategies

This section introduces a number of personalization models focused on providing users with information that match their interest. The personalization models utilize the user (and group) models (see Section 4.2) for adapting information to heterogeneous individuals. This adaptation is represented by a recommendation, i.e. a list of items ordered by relevancy or tag clouds, a cluster of terms that helps users to navigate through a collection of documents.

4.4.1 Recommendations based on Signal Definitions

As shown in Section 2, users create signal definitions to perform surveillance of specific diseases in determined locations. These inputs can be understood as fundamental source for personalization once they expose the user’s interest. In addition to the personalization model that relies on the user’s tagging as shown in Section 4.3.1, we also proposed a method that utilizes signal definition information to provide personalized recommendations. The limitation of this method is that the recommendations are constrained solely by the parameters (e.g., location, medical conditions) and related information that may be still important are not retrieved. The complete details of this model can be seen in *Section 2.2 of M-Eco Deliverable D5.1* [60].

4.4.2 Tag-Based Recommendations

Tags are potential indicators of user preference. For instance, a medical expert that has exhaustively assigned the tag “*swine flu*” to the documents he evaluates, seems to be interested in that disease. Therefore, this knowledge can be utilized to filter out irrelevant recommendations unrelated to “*swine flu*”. For recommending items to the user, we compare his tags, i.e. tags assigned by him to his documents of interest against the tags assigned to candidate and unknown documents. The comparison is realized by the cosine similarity of two tag vectors, one corresponding the user’s tag vector and the other corresponding to the document’s tag vector. The documents with highest similarity to the user’s tag profile is then selected to be recommended.

Worth mentioning that although the tag-based recommendation component appears as a single component in Figure 7, it is not utilized as a stand alone application. Instead, this component is adapted to some extent and reused by other components such as the *Semantic Enhanced Tag-Based* component, *Tag Cloud* component, *Tag Neighbors* component and *Spectral Clustering of Tag Neighbors* component.

4.4.3 Recommendations based on Locations

In Section 4.2.3 we present the model for user location preferences. It is built from the set of user locations defined in his or her signal definitions and from similar locations. Location similarity is computed as a function of two factors:

- Political hierarchy (e.g., city, state or country level); and,

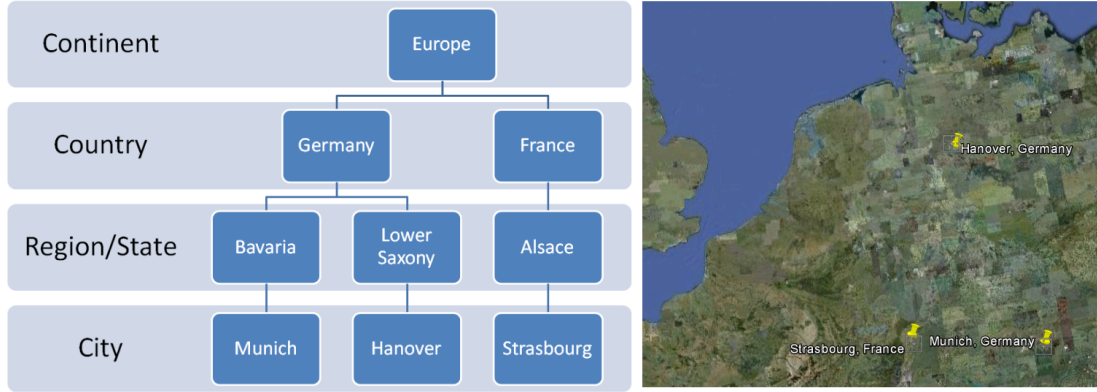


Figure 11: Example of political hierarchy and corresponding locations on a map.

- Distance and population.

Political hierarchy is used to compute the political distance. This distance is a function that closely associates how "connected" two cities are based on their hierarchy. This hierarchy is fixed and contains the following levels: city, state (or region), country, and continent. This is then used to find the common ancestor between two locations. For example, Munich, a city in the state of Bavaria, and Hanover, a city in Lower Saxony have as their common ancestor the country (Germany). On the other hand, Strasbourg, a city in France, and Munich have as their common ancestor the continent (Europe). Figure 11 illustrates the example. This intuition assumes a distance d_h of 10^n units where n is the number of levels above the city level two locations are in the hierarchy. For example, Munich and Hanover have a political distance of 100 units ($d_h = 10^2 = 100$), and Strasbourg and Munich have $d_h = 10^3 = 1000$. We assume an exponential scale given the much higher importance that should be given to more politically nearby locations.

The motivation for this distance is the fact that two locations within the same political hierarchy are more likely to be of interest to a health official which is restricted to actions within his or her administrative region (e.g., a state or country). For example, a health official concerned with outbreaks in Lower Saxony may not have any authority or the responsibility to act on similar events occurring in the Netherlands. The political distance, then, increases the importance of cities within Lower Saxony instead of cities in other regions or countries even though the actual geographic distances of the latter might be lower.

In addition to the political distance, the actual distance and the population of the locations are used to compute the proximity distance d_p . This distance is initially computed as the geodesic distance, d_g , i.e. the shortest distance between two points on earth. It is then weighted according to two factors: population size and airport connections

The population size is used to determine the probability of travels between two locations. The assumption is that the more populated a location, the more likely people are to travel there. Therefore, if two locations have the same geodesic distance from a third location, the one with a bigger population will have a higher proximity distance score. The simple probability t of

traveling to a location is computed as:

$$t = \frac{p_l}{\sum_{i=1}^{|L|} p_i}, \quad (4)$$

where p_l is the population of location l , $|L|$ is total number of locations available, and p_i is the population of the i th location.

Port and airport connections, when available, influence how well two locations are connected despite their geodesic distance. For example, the cities of London and Paris could have a higher proximity distance score than Paris and Strasbourg, despite their similar geodesic distances, because of the number of direct flight connections per day between the first two cities. The number influences the proximity distance because it increases the probability of higher flow of people between two locations.

The initial probability t above, given now as the probability t_{l1l2} between locations $l1$ and $l2$, is then modified to account for the number of flight connections per day between them:

$$t_{l1l2} = t + \frac{c_{l1l2}}{|C|}, \quad (5)$$

where c_{l1l2} is the number of flight connections between locations $l1$ and $l2$, and $|C|$ is the total number of flight connections for all locations.

t_{l1l2} is then used to weigh the original geodesic distance, d_g , and compute the proximity distance score as:

$$d_p = d_g \times t_{l1l2} \quad (6)$$

To compute the final location similarity score (LSS), we determine empirically the relationship between the political, d_h , and proximity, d_p , distance. We compute their values pairwise for a set of 160 locations related to the EHEC outbreak in Germany described in Section 2.3. Analyzing the relationship between the distances we find a small significant correlation ($r = 0.45$). This means that there is a relationship between the proximity and political distances.

This is the case because most of the lower political distances ($d_h = 10$ or $d_h = 100$) are related to lower proximity distances $d_h < 1500$, i.e. cities nearby one another tend to be within the same region or country. The exceptions are mostly locations in countries with large territorial areas such as Russia and the United States (US). The correlation is not higher because the locations with higher political distances ($d_h = 1000$), i.e. between those locations that are not part of the same region or country, have varied proximity distances. For example, most locations in Denmark have low proximity distances to locations in the north of Germany but all of their political distances are set to 1000. On the other hand, all locations in the US also have political distance of 1000 with Germany but much higher proximity distances.

Since only a handful of pairs of locations have lower political distances, we consider only those pairs with $d_h = 10$ to weigh up the final location similarity score. The locations with $d_h = 100$ are unaffected and locations with $d_h = 1000$ or higher are penalized. With this assumption, we define the location similarity score (LSS) as:

$$LSS = 10 \times d_h^{-0.5} \times d_p \quad (7)$$

LSS can be pre-computed for all pairs of locations available on M-Eco. Once this process is completed, the score is normalized to range from 0 to 1.

Recommendations based on locations. With Location Similarity Scores pre-computed, we use them in cosine similarity measures to find out items that are of related locations. This takes into account the set L_u of location preferences of the user, as discussed in Section 4.2.3. For each location in the set, we build a vector, V_l , of location similarity scores, where:

- The actual locations the user chose in his or her signal definitions receive a score of 1.
- Locations in L_u that are similar to the user chosen locations receive the highest corresponding score from LSS in relation to the chosen locations.

The same process is repeated for items available in M-Eco that match the medical conditions specified by the user in his or her signal definitions. Then, for each item i we build the vector V_i of location similarity scores related to the original locations found in i . Finally, once the vectors are built, we use the cosine similarity measure between V_l and V_i to determine the similarity between the user location preferences and the location of an item i . This final similarity score is then used as one of the factors by the group recommendation model as described in Section 4.4.4. Further, the evaluation of the location-based recommendations are shown in Section 5.6.

4.4.4 Group Recommendations

In Section 4.2.2 we presented our group preference model. It is built from the individual user interactions with published items. The group recommendation process takes into account the group preferences based on the actions of its members to decide which items to publish next[36]. Such recommendations can be useful when individuals are working together in areas that require relevant up-to-date news information to support decisions. One of the main challenges we attempt to address is to make recommendations to groups of individuals maximizing the overall group satisfaction and, at the same time, keeping individual dissatisfactions low.

To test our recommendation model, we used Twitter, a microblogging service, as a platform to recommend short messages to a group of users following such a service account. The scenario with Twitter was previously described in Section 2.5. Messages published on the service are references to web pages together with their respective links for the users to read the full content if interested. To this end, the system built to publish the messages is composed of three main modules:

- The *Source Manager* retrieves news articles from external information sources according to a predefined schedule policy. The data is processed so that the title, link and terms are extracted from the text and saved in a local database. This information is used to produce the short messages to be published, containing the title and link to the original article. The terms extracted from the article are indexed in order to be used as part of the recommendation calculus.
- The *Service Monitor* observes the users' activities in the microblog account and gathers the messages that are forwarded or marked as favorite. Such user data is obtained through an API ⁵ and utilized to build up the group profile.
- The *Recommender Module* builds the group preference based on the Group Strategy chosen. It compares the (new) candidate messages retrieved by the *Source Manager* with the group preference elicited. Then, the candidate messages more likely to be accepted by the

⁵In the particular case of Twitter, we use the Twitter4J API (<http://twitter4j.org>)

group are recommended, i.e. published in the microblog account. The recommendation model is depicted in the following subsections.

Generating recommendations. The Recommender Module periodically updates the group preferences as described in Section 4.2.2. Once an updated group profile is available, the score of a candidate message m to be recommended to the group is calculated as:

$$Grec(m, GP, IM) = \alpha(sim(m, GP)) - (1 - \alpha)(sim(m, IM)), \quad (8)$$

where the message m with the highest $Grec$ score gets selected to be published. That is, the message m selected has a high similarity with the group model preference GP , and a low similarity with the set of ignored messages IM . Additionally, we utilize a weighting factor $\alpha \in [0, 1]$ to calibrate the importance of GP over IM and vice-versa. This factor is needed to reflect more properly the current group activity in the microblog account where the messages are being published. For instance, if the group activity is low and many messages are ignored, then IM becomes more important than GP . As a consequence, the α value is suggested to be low, i.e. close to zero.

The calculation of similarities in $Grec$ is given by the cosine similarity measure. Such measure is used to compute $sim(V_m, V_{IM})$ and $sim(V_m, V_{GP})$, where V_m , V_{IM} and V_{GP} are the corresponding term vectors of message m , ignored messages in IM , and preferred messages in GP , respectively. The element values of V_m and V_{IM} are computed by the $TF \times IDF$ formula, where TF is the term frequency of a term t in the corresponding content of a particular news article the message m links to, and IDF is the inverse document frequency of the term t , based on the total number of news articles where term t occurs. Note that the corresponding content linked to the message is not a short tweet message. It is the content of the whole article linked to the message, external to twitter, and therefore long enough to use the $TF \times IDF$ measure. Such measure would not be well enough suited if we had only considered the tweet message itself. This way we are also more secure about the meaning of the message itself.

To calculate the element values of V_{GP} we propose a new weighting scheme to prioritize messages the users interacted more with and to account for their evolving preferences. In this scheme we modify the $TF \times IDF$ formula to introduce the following factors:

- *utility value* – We credit the importance of a message by the amount of interactions the group members have with it;
- *time decay* – We assume that the interactions that took place more recently represent a higher indicator of the current preferences of the group. This is also used to reduce the recommendation of similar content for a long period of time, counterbalancing the utility value.

Thus, for each pair $[message, value]$ in GP , each of the message's terms t is weighed as follows:

$$termvalue(t) = (tf \times \log(N/n)) \frac{value}{tm}, \quad (9)$$

where tf is the term frequency of the term in the news article, N is the total number of news articles in GP , n is total number of news articles in GP that contains the term t , $value$ is the utility value, and tm is the delta time from the date of the last group interaction with

the message and current date. An element value of V_{GP} is then calculated as the sum of term values of a term t in all news articles where it appears.

As stated before, terms are extracted from the news articles. Their $TF \times IDF$ value is then weighed according to the interaction of the group in the message published on the microblogging account and with a time decay value. Worth mentioning that this message is composed by a highlight text, links to the news article, and the evaluation link. When comparing our approach with others from the literature, we weigh the term's $TF \times IDF$ value with the explicit ratings given by the users in the group, according to the respective strategy. For example, in the Least Misery Strategy, the $TF \times IDF$ value is weighed by the highest minimum rating given by the group on that particular article. In all cases, the weights are normalized to ensure compatibility of the results. The next section the tag cloud model to help users navigating through documents.

4.4.5 Personalized Tag Cloud Adaptation and Navigation

In the following section, we present technical solutions for a generation of personalized tag cloud which was introduced and motivated in Section 2.6. This visual interface provides an alternative way of retrieving and browsing a large amount of documents related to signals that match a particular signal definition. The tag cloud is a summary of keywords that represent a particular signal which has been detected in the documents. It provides an overall picture about a considered signal. The goal is to provide a possibility for epidemic prediction experts to validate signals through exploration of the tag cloud. Tags of the tag cloud can be treated as extended symptoms of the signal. It gives an easy access to the documents under generated signals by simple clicking on particular tags. In this section, we describe a personalized tag cloud model with a specification of all components that are involved in a tag cloud generation.

The process of tag cloud generation consists of the following six steps: i) Retrieve all tags assigned to the documents that are related to a given signal(s), ii) apply syntactical pre-clustering of tags so that terms with typos, singular and plural of tags are grouped into the same clusters. Each cluster is represented with the most frequent tag, iii) syntactically grouped tag space is clustered based on the co-occurrences of tag pairs, iv) tags are selected from generated clusters such that tags with higher coverage are preferred, v) tags from the same cluster are located close to each other and share the same color. The size of tags is predetermined by the corresponding coverage value, vi) if a user's profile is available, a generated tag cloud is adjusted such that it contains mainly tags from the clusters related to the user's interests. The following sections provide further details of some steps above.

Syntactical Pre-clustering of Tags. Syntactical pre-clustering filters out tags with typographical misspellings (E.colli, E.coly) unnecessary plural and singular forms of the same tag (cucumber, cucumbers) and also compounded tag from two different terms connected with some separator (E.coli, E-coli, E coli). These useless tags would occupy a tag cloud's space and in consequence it would result in the semantically redundant tags in the cloud. Syntactical pre-clustering affects a structure of the generated tag cloud in the sense that depicted tags are semantically more diverse.

Levenshtein distance is computed for each tag pair from the initial tag space. The distance between two tags measures the number of required changes (substitution, insertion and deletion of a character are allowed operations) to transform one tag into another. We justify its use because it attains significantly better results than Hamming distance as shown in [16]. Once,

an edit distance is calculated - tag space is divided into clusters. Each group contains only tags where Levenhstein distance is equal or lower than a defined threshold (a number of maximum changes to transform a tag from the tag pair into a second tag). Then, the most frequent tag for each cluster is selected and is used in all further computations and represents all other tags from a considered cluster (E.coli represents E-coli, E coli, E.colli and E.coly).

Tags Clustering. After the syntactical care, a refined tag space is clustered based on the tag pairs co-occurrences. The goal is to group semantically similar tags into clusters which results into the following benefits:

- Synonymous tags appear together in the tag cloud and are displayed with the same color. This presentation structure differs from the most common visualization of tag clouds where tags are alphabetically sorted. It allows to differentiate main topics in the tag cloud and also users can perceive and notice semantic relations between tags in neighbourhood [22]. The latter helps to understand connections between tags, for example, tags cucumber and Spain are hardly interpretable in alphabetically sorted tag cloud. However, if they are depicted together with the tag E.coli a user can easily assume that these tags are related to E.coli outbreak.
- Clustered tag space is utilized as the main source when generating personalized tag cloud. Clusters that contain user's tags are identified and other tags from these selected clusters are utilized for building a personalized tag cloud.
- A tag cloud becomes more diverse as tags are selected from different clusters in comparison to alphabetically sorted tag cloud where selection is performed based on the tag frequency.

We present three different clustering techniques – the first two proposed approaches cluster tags according to their co-occurrence based similarities, K-means algorithm considers each tag from a tag space as feature vector. These techniques were proposed and described in [39].

Correlated Feature Hashing. We propose to reduce a tag space with hashing function that is similar to the proposed technique in [4] where authors successfully reduced dictionary size by utilizing hashing. The idea is to share and group tags with the similar meaning.

We sort the tags used within the system according to the frequency of usage such that t_1 is the most frequent tag and t_T is the least frequent. For each tag $t_i \in 1, \dots, T$ is calculated *DICE* coefficient with respect to each tag $t_j \in 1, \dots, K$ among the top K most frequent tags. The *DICE* coefficient is defined as:

$$\text{DICE}(t_i, t_j) = \frac{2 \cdot \text{cocr}(t_i, t_j)}{\text{ocr}(t_i) + \text{ocr}(t_j)} \quad (10)$$

where $\text{cocr}(t_i, t_j)$ denotes the number of co-occurrences for tags t_i and t_j , $\text{ocr}(t_i)$ and $\text{ocr}(t_j)$ is the total number of tag t_i , t_j assignments respectively. For each tag t_i , we sort the K scores in descending order such that $S_p(t_i) \in 1, \dots, K$ represents the tag of the p -th largest *DICE* score $\text{DICE}(t_i, S_p(t_i))$. We can then use hash kernel approximation defined as:

$$\bar{\Phi}_{t_j}(x) = \sum_{t_i \in T: h(t_i)=t_j} \Phi_{t_i}(x) \quad (11)$$

and given by a hash function:

$$h(t_i) = S_1(t_i) \quad (12)$$

The described approach is replacing each tag t_i with the tag $S_1(t_i)$. Obviously, we have reduced tag space from all T tags to the K most frequent tags.

Spectral K-means clustering. In the second approach we utilize Spectral K-means clustering technique. Firstly, we encode tag relations into the affinity matrix W , such that $w_{i,j}$ entry represents affinity between tag t_i and tag t_j . The similarity matrix can be also interpreted as undirected weighted graph G where tags represent nodes and weights are expressed as similarities between given tags. We use *DICE* similarity (10). Once the similarity matrix W is created, we then proceed to find (sub) clusters of tags that address the same topic. To obtain clusters, we rely on a spectral clustering algorithm which input is the undirected weighted graph G . The spectral clustering algorithm partitions the graph G based on its spectral decomposition into sub graphs. In order to run the spectral clustering, we perform the following steps:

1. We build the Laplacian matrix $L = D^{-1/2}WD^{-1/2}$ derived from the affinity matrix W . The D is $n \times n$ diagonal matrix whose (i, i) - th element is the sum of W 's i - th row, in other words it is the degree of a given node i - sum of all weights corresponding to the edges that are connected to a given node i . The Laplacian matrix L is symmetric and has identical size as affinity matrix W .
2. We compute the k largest eigenvectors of L , these obtained top k eigenvectors are used as columns to create a new matrix $U \in \mathcal{R}^{n \times k}$. We consider each row of U as a point in \mathcal{R}^k , hence we can apply standard K-means algorithm to cluster these points into k clusters.
3. Finally, we map original node i to the cluster j if and only if row i from matrix U belongs to the same cluster j . We obtained disjoint groups of similar and related tags and we are able to generate a tag cloud.

Two described clustering techniques utilize our own hybrid similarity measure which combines *DICE* coefficient of a tag pair (t_i, t_j) and semantic relationship retrieved from WordNet dictionary.

$$HybridSimilarity(t_i, t_j) = 0.5 \cdot DICE(t_i, t_j) + 0.5 \cdot WordnetRelation(t_i, t_j) \quad (13)$$

where a function $WordnetRelation(t_i, t_j)$ is defined as follows:

$$WordnetRelation(t_i, t_j) = \begin{cases} 1, & \text{exists an relation for } (t_i, t_j) \\ 0, & \text{no association between } (t_i, t_j) \end{cases} \quad (14)$$

K-means Algorithm. The following clustering technique differs from the previous in such a way that each tag is expressed in n -dimensional vector space where i -th dimension corresponds to the i -th item res_i (in a similar way as in [46, 31]). We denote $T = \{t_1, t_2 \dots, t_{|T|}\}$ as the set of all distinct tags that are clustered and $R = \{res_1, res_2 \dots res_n\}$ the set of all items that are tagged with tags from T . Let $f(t, res_i)$ be equal to a frequency of a tag t assignments to item res_i otherwise it is equal to 0. Then, the vector representation of tag t is:

$$t = (f(t, res_1), f(t, res_2), \dots, f(t, res_n)) \quad (15)$$

Once, tags from T are expressed as n -dimensional vectors, we proceed with the cluster analysis. The K-means is a simple well known clustering technique that groups objects from a given set into k clusters (given a priori). The algorithm starts with generating k random centroids. Then for each object from a dataset, i.e. for a tag from T , the nearest centroid is found. Thus the given tag is associated with a particular centroid. When all tags are processed, centroids have to be recomputed such that new centroid is the mean value of the vectors for the given cluster. Again for all objects the nearest centroids are identified and objects are clustered with them. This process repeats until locations of centroids do not change. The clustering of a tag space with the K-means algorithm is computed as follows:

1. Each tag from a tag space T is expressed as n -dimensional vector. According to the size of the tag space and user requirements an amount of clusters is set to k .
2. It randomly places k centroids such that a distance from each other is maximized.
3. Each tag from the tag space is bound to the nearest centroid.
4. New centroids are computed as the mean value of tags vectors grouped with a given centroid. It continues with the step 3, until new centroids are identical with the centroids from the previous iteration.

We obtained k disjoint clusters of tags so we can proceed with the tag cloud visualization. The results of K-means algorithm depend on used distance measure - we exploit only Cosine distance as it attains the best results [39].

Each from the presented techniques can be utilized in a process of tag cloud generation. Spectral k-means generates the most reasonable clusters however the time performance is not feasible. However, Correlated Feature Hashing attains comparable results and time complexity is significantly reduced. K-means is an alternative way of clustering for cases when tags in the tag space do not co-occur often. Because of the afore-mentioned reasons, we utilize Correlated Feature Hashing as the tag spaces for majority of signals in M-Eco project are large enough and tags co-occur a lot.

Tag Cloud Generation. Finally, clustered tag space is depicted in the tag cloud. Semantically related tags from the same cluster are displayed with the same color which is specific for each cluster. Such tags are also located near each other and it allows to explore tags in more convenient way. Location and particular color of tags from the identical cluster results into a tag cloud which is semantically structured and as was shown in [22].

Tag space usually contains thousands of different tags therefore, there is a need to select only the most significant ones as the amount of terms depicted in the tag cloud is limited. The most common approach is to select tags according to their frequency of use. However, we select tags according to the following metric:

$$Coverage(t) = \frac{|D_t|}{|D_a|}, \quad (16)$$

where $|D_t|$ is the number of documents assigned with a tag t and $|D_a|$ is the number of all documents that are considered during a tag cloud generation process. The metric ranges between 0 and 1. When a coverage for a particular tag t is close to 1 majority of considered

documents was annotated with a tag t . We utilize this metric during the selection process to maximize number of documents that can be accessed directly by exploring a tag cloud.

Personalized tag cloud is an adapted version of the above-mentioned general tag cloud model. User's preferences are incorporated into the tag cloud such that tags related to user's tags are preferred over others. The selection of similar tags is performed by retrieving tags from the clusters that contain at least one of the user's tags. Tag cloud is by default generated from the tags of all documents available in the system. However, the tag cloud can be build also according to the following parameters:

- signal - a tag cloud is generated only from a specific set of documents related to a given signal.
- language - a tag cloud is generated only from a specific set of documents written in a given language.
- location - a tag cloud is generated only from a specific set of documents related to a given location.

4.4.6 Generative Model of User Taggings

A generative model of user taggings is a supplementary component for a tag cloud generation (Section 4.4.5) and tag-based recommendation models (Sections 4.4.2, 4.3.1). The problem of social tagging systems is the aggregation of a vast number of documents that should be tagged by users in order to provide a tag-based services. Hence, user tagging data is usually scarce, it gets difficult to learn user preferences and therefore to generate proper tag-based recommendations or personalized tag clouds. In order to overcome such an inconvenience, we propose a generative model of taggings for medical documents based on the analysis of the tag representativeness distribution in related social bookmarking systems. The objective is to facilitate a tagging process of documents with the automatic generative model that produces taggings for each document in the system.

The proposed probabilistic model generates a tagging annotations based on a computed tag representativeness distribution from a different folksonomy. A tag representativeness is the number of occurrences of a considered tag within a given document (e.g., a tag *car* has been assigned to a document *doc1* and a text of *doc1* contains term *car* 2 times therefore, a tag representativeness for the tag *car* and document *doc1* is 2). The motivation is to simulate tagging patterns of users so that produced tags from the generative model resemble as much as possible to the users tags.

We consider a folksonomy data from Delicious service. The distribution was computed for 2000 most frequently tagged documents and only top-25 tags with the highest tag representativeness are considered. Once, an average distribution for all considered documents is computed, the model proceeds with a tag extraction process from M-Eco documents. All terms relevant to medical topics are preferred over other terms. For a considered document there are identified terms that are nouns then all such extracted terms are matched against a medical dictionary that contains ~ 67000 medical terms. The medical phrases and words are preferred over non-medical and according to their term frequencies are sorted in the descending order. These ordered list of 25 terms is used as the sample space for a tag generation process. We consider only 25 most representative tags as other less representative tags have insignificant participation in the general distribution. Once, a list of extracted terms is sorted we can proceed with the generation of tags for a particular document. A generative model is based on a

roulette wheel sampler which based on the considered probability distribution generates randomly a set of tags. The model generates tags relevant to medical domains which is important in the M-Eco system. It allows to generate and users' taggings when real users are not available and in a such way avoid a cold start problem.

The main limitation of the proposed model is a restriction of generating only representative tags. It does not generate tags that do not occur in the text of a considered document.

5 Evaluation

In this section, we present the experimental evaluations of the personalization models (see Section 4) that address the problems identified in Section 2. The experimental evaluation aims at testing whether our proposed approach performs as expected so that we can claim our methods as contributions to solve or lessening the problems in the motivation of this work. The evaluations are represented by *user study*, in which participants assess the performance our developments (recommendations and tag clouds); *interviews and group discussions*, in which the participants answer pre-defined questions or provide feedback of our methods; and *controlled experiments*, in which public medical datasets are utilized for the assessment of our methods.

5.1 User Evaluation of Recommendation Presentation and Feedback Options

The scenario proposed for our evaluation used real, mainly consisted of signals related to mentions of the H1N1 virus, but we did add sparse signals on two other diseases, E. Coli and the common flu [37]. In total, we had 138 signals and 419 documents belonging to one of the signals in the test set. In this scenario, we wanted to test, in general, how the recommendation component would perform and, more specifically, whether it would filter out *E. Coli*, recommending mostly those related to H1N1 for the users who set their signal definitions to this disease and some related to the common flu.

In addition to the data available, participants of the evaluation were requested to create signal definitions according to their interest (and not necessarily about H1N1). In the prototype presented, a signal definition represented 2 sets of features linked to the participant's interest: medical conditions and locations. They could also provide a title and a description to the signal definition. The information provided by the participants in their respective signal definitions was then used to generate personalized recommendations to them. These recommendations were then assessed by the participants according to the following criteria: *Relevance*, *Correctness* and *Representativeness*.

Relevance was defined subjectively as whether the recommendation is useful or can be directly applied to the user's work. A recommended signal could only be marked as relevant or not relevant. We used the relevance to compute the precision of our recommended signals. *Precision*, in our context, measures the probability with which a recommended item (i.e. a signal) is indeed relevant to the user, and is defined as follows:

$$Precision = \frac{R}{S}, \quad (17)$$

where R is total number of recommendations marked as relevant by the user and S is total number of signals that were recommended. *Correctness* was defined as how well the

recommended signals match the user preference with regard to the signal definitions set by him. *Representativeness* was defined as how well the set of recommended documents in the recommended signal represents or matches that signal. In other words, it indicates how good is the degree of belonging of the set of recommended documents to the recommended signal.

To generate recommendations, we opted for a small number of participants in order to restrict it only to health surveillance experts. They were representatives of the World Health Organization (WHO), the French Institut de veille sanitaire (Sanitary Surveillance Institute - INVS), the European Centre for Disease Prevention and Control (ECDC) and the Mekong Basin Disease Surveillance (MBDS) consortium. 8 participants joined the evaluation as users of a prototype of the M-Eco, an event-based surveillance system.

Once logged in the system, the users were asked to create signal definitions according to their interest, choosing from a set of medical conditions and locations, and providing a title and a description for the signal definition. The system then produced the recommendations of existing signals and documents to the users based on their signal definitions. In total, for each user, the 5 best ranked signals matching the user interests defined in his signal definitions were shown in the prototype as recommended signals. Each recommended signal had a set of the 5 best ranked documents matching the signal features (i.e. medical conditions and locations) and the user's signal definitions. These documents were shown in the prototype as recommended documents of the selected recommended signal. The last step was the actual evaluation of the recommendations. In the system, users were asked to mark the recommended signals as relevant or not, and to also grade them in terms of correctness and representativeness in a four-point Likert scale. Since the users were free to use the system as they wanted, not all of them evaluated all recommended signals. In total, 31 recommended signals out of 40 generated were evaluated.

The ratings given to each criterion were complemented by a focus group discussion where the participants provided feedback on their experience with the prototype. A focus group we set to moderate the discussion on a specific set of topics where 6-12 people participate. The small group allows for a shared understanding of specific topics while allowing individual differences to be voiced. Questions are open-ended but pre-defined. A moderator ensures that the discussions stays within the proposed topics. Advantages of focus groups include flexibility, due to its open format. It also benefits from the opportunity of a direct interaction with the participants, allowing for clarifications and further exploration of specific topics. In that sense, other participants can build on other's comments, enriching the discussion. In this evaluation, the questions posed in the discussion were:

1. How would recommendations help you deal with the excess of information you might receive?
2. How would you like to evaluate the recommendations you receive? More specifically, are the options currently provided (relevance, correctness and representativeness) appropriate?
3. How should the recommendations be placed in the system?

Discussions took approximately one hour and a half. We did not seek exact answers to these questions from the participants. Rather, we opted to use them as triggers to topics that could help us understand how health surveillance experts perceive the benefits of recommendations. Our goal was to learn from them how to present recommendations in surveillance systems and how to offer them alternatives of giving feedback to improve recommendations over time.

More specifically, question 1 aimed at assessing whether recommendations can indeed help users of surveillance system and whether they find it necessary/important. Question 2, at the other end, has the objective of identifying how the users would like to evaluate recommendations received. For instance, more feedback can help to improve future recommendations but, at the same time, to prevent more users to do so due to time constraints. Finally, question 3 focuses on stimulating discussions on the GUI, in order to assess how recommendations should be provided in a surveillance system.

The next section presents first the results of the evaluation of the generated recommendations. It is followed by the feedback given by the participants in the focus group.

5.1.1 Recommendation Results

Results showed that the recommendations were relevant for the 5 users who did set up at least one of their signal definitions to H1N1 but practically irrelevant to the other 3. This confirmed our expectation, given the dataset focused on H1N1, that the recommendation component would filter out recommendations of other, non-related, diseases. Users that did set up signal definitions to H1N1 also gave high grades to the correctness of the recommended signals and to the representativeness of their respective recommended documents. Table 3 summarizes the results.

Measure	H1N1	Non-H1N1
Precision	0.81	0.27
Correctness	3.25	1.53
Representativeness	3.37	1.33

Table 3: Summary of the results in terms of precision and mean of correctness and representativeness for the ratings given to recommendations matching H1N1 and non-H1N1 signal definitions

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.4060	0.0836	-4.86	>0.001
Correctness	0.2383	0.0726	3.28	>0.01
Representativeness	0.1312	0.0727	1.80	>0.05
Residual standard error: 0.2126 on 28 degrees of freedom				
Multiple R-squared: 0.8366, Adjusted R-squared: 0.8249				
F-statistic: 71.68 on 2 and 28 DF, p-value: 9.67e-12				

Table 4: Linear regression model: Relevance \sim Correctness + Representativeness

For the recommended signals matching H1N1 signal definitions, 13 recommended signals were marked as relevant out of 16, thus achieving a precision of 0.81. The 3 non-relevant recommendations were related to the common flu, instead of H1N1. This could indicate that users are not necessarily interested in related types of diseases. By contrast, for the other recommended signals, not related to H1N1, only 4 recommended signals were marked as relevant out of 15 possible, a precision of 0.27.

Similarly, recommended signals matching H1N1 signal definitions received high grades in terms of correctness (mean = 3.25) and representativeness (mean = 3.37). The recommended signals matching other signal definitions had much lower grades for both correctness (mean = 1.53) and representativeness (mean = 1.33). Note also that there is a linear relationship between relevance (an implemented feature of M-Eco) and the grades of correctness and representativeness that were only included as part of this evaluation. This indicates that relevance could already be capturing other aspects of the recommendation, thus eliminating the need of more measurements in future large-scale experiments. The relationship could also help explain what it means for a recommendation to be relevant: that is, to be also correct and to have a representative set of documents attached to it. Table 4 shows the results.

5.1.2 Group Discussion

After the evaluation of the recommendations using the prototype, the participants were invited to engage in a focus group discussion where they provided feedback on their experience with the evaluation and on their wishes regarding recommendations in the M-Eco system. Discussion were triggered by the set of questions posed previously but moved freely from there once the participants engaged in conversations.

In the first question, participants were asked whether recommendations can indeed help users of surveillance system and whether they find them necessary/important. In general, all participants agreed that the system advises users about something that they are missing or have not considered to define in the rules. That is, recommendations should help them discover topics related but not an exact match to their current preferences. This observation is in contrast with the ratings received on the recommendations about common flu, as described in previous section. In that regard, they mention that the system should also be flexible enough in order to learn from the user feedback but also to “forget”, as one of them said, so that the users’ changing preferences are updated in the system.

In addition, the participants also agreed that recommendations should assist users in overcoming the overload of signals they might receive. However, the system should avoid turning the recommendations into a problem themselves, by overloading the users with them. According to one of them, *“it is normal that similar alerts are set up by different users”* so recommendations should avoid focusing too much on many similar signals. An alternative presented to overcome this issue was to use recommendations as background information. In this case, information given to a particular user would not be about an actual event, but of previously identified events that can also be of potential interest. They noted, however, that this would be a complementary rather than preliminary type of recommendation. For instance, if the user is willing to spend time, then it would be useful to refer to historical and contextual information about a current event in order to support decision making.

The second question asked how the participants would like to evaluate the recommendations. Most participants agreed with the evaluation options provided currently in the GUI but one in particular suggested that the same options should be available also for the documents of a particular signal, not only the signals. In addition, another participant suggested that users should be allowed to tell their preferred recommendations by re-ranking the list provided. According to him, this could be complemented by a *“I don’t want to see this anymore”* option.

This option is similar to a “dislike” or “irrelevant” button as others pointed out. This would then complement the “relevant” button currently available, letting the system know that the user is not interested in recommendations from a particular topic. The “dislike” button could

also help avoid the problem of recommending very similar topics as pointed out by one of the participants.

These ratings could then be complemented by an explanation given by the user. The idea discussed is that relevance is relative and a "yes/no" evaluation may not represent the proper interests of the user. A free-text explanation then could be used as a complementary evaluation tool. Similarly, the relevance is also relative in terms of time. As mentioned before, the system should be able to learn and also forget previously defined user preferences, so that it can adapt to changing behavior. To the participants, a rating given to a recommendation can never be considered as an absolute and definitive statement of user's preferences.

The third and last question focused on the elements of the GUI related to the recommendations. First, they considered that the term "recommendation" or "recommended signal" (as it was displayed in the GUI) not very appropriate. Instead, separated areas could be provided with proper designations of what are the recommendations about. The examples given were "Related signals" and "Other signals that might interest you".

In fact, one of the participants' main suggestions was that an explanation should be given to the user about the origin of a recommendation. That is, they would like to know the reason why each recommendation was generated in order to give an accurate feedback to it. For instance, by knowing that a particular recommendation came from a specific set of rules, a user could mark it as irrelevant because he is not interested in more information about that particular topic.

5.2 Evaluation of Spectral Clustering of Tag Neighbors for Recommendation

The evaluation of the spectral clustering of tag neighbors for recommendation are performed on the "M-Eco system" dataset containing 43 different users, 133 documents and 723 tags with 2794 tagging annotations. The setting of the dataset is that each user and document has at least 5 annotated tags. Following the traditional protocol which chooses 20-30% data for the testing data, the data was divided 75% as the training data and the rest 25% as the testing data. The goal of such experiments is to show that the document recommendations derived by using our proposed approach would result in an improvement of document recommendation performance.

Modularity Metric. Before we validate the performance of this proposed recommendation approach, we first aim to investigate the impact of choosing different numbers of clusters in order to get the optimal number of clusters. Here we use the metric of modularity to evaluate the experiments. Modularity is originally proposed to assess the division quality of a network. The modularity of a particular division of a network is calculated based on the differences between the actual number of edges within a community in the division and the expected number of such edges if they were placed randomly.

Consider a particular division of a network into k communities. We can define a $k \times k$ symmetric matrix SM whose element sm_{ij} is the fraction of all edges in the network that link vertices in community p to vertices in community q . The similarity of smC_{pq} between the two

clusters C_p and C_q as:

$$smC_{pq} = \frac{\sum_{c_p \in C_p} \sum_{c_q \in C_q} c_{pq}}{\sum_{c_p \in C} \sum_{c_q \in C} c_{pq}}, p, q = 1, 2, \dots, m, \quad (18)$$

where c_{pq} is the element in the similarity matrix SM . When $p=q$, the smC_{pq} is the similarity between the elements inside the clusters, while $p \neq q$, the smC_{pq} is the similarity between the cluster C_p and the cluster C_q . So the condition of a high quality cluster is $\arg \max_p (\sum_p smC_{pp})$ and $\arg \min_{p,q} (\sum_{p,q} smC_{pq}), p \neq q, p, q = 1, 2, \dots, m$. Summing over all pairs of vertices in the same group, the modularity, denoted Q , is given by:

$$Q = \sum_{p=1}^m [smC_{pp} - (\sum_{q=1}^m smC_{pq})^2] = TrSM - \|SM^2\|, \quad (19)$$

where the m is the amount of clusters. The trace of this matrix $TrSM = \sum_{p=1}^m smC_{pp}$ gives the fraction of edges in the network that connect vertices in the same community. This quantity measures the fraction of the edges in the network that connects vertices of the same type minus the expected value of the same quantity in a network with the same community divisions. Here we compare the result of Q values by using Spectral Clustering, Single Linkage Clustering and Random Clustering. The result below is based on the average result of executing the same experiment ten times over the same dataset.

5.2.1 Precision Evaluation and Evaluation Results

In the experiment, we compared the precision for the recommended documents to the individual user. The traditional way is to calculate the similarity between the user profile and the document profile in the tag vector, and the system will recommend the N documents to the user according to the top- N similarity values. Another approach is to calculate such similarity based on the naive tag neighbor expansion by collaborative filtering approach. However, the disadvantage is that such tag neighbors include many noisy tags, degrading the precision of recommendation. Our proposed approach is to filter out the noisy tags by utilizing the clustering.

We will calculate the precision in such step: The top- N documents will be recommended to the user by ranking the similarity values derived with various approaches, the recommended documents are compared with the existing documents in the test data. If there are k documents appeared out of the N recommended documents in the test data, the existed number of test documents for each user is E_i , the precision for the individual user is defined as t :

$$t = \frac{K_i}{N} \times 100\% \quad (20)$$

When N is bigger than E_i , we consider the precision as:

$$t = \frac{K_i}{E} \times 100\% \quad (21)$$



Figure 12: Optimal weight of λ .

To assess the setting of λ , we average the precision over top-N recommendations, t

$$t = \sum_{i=1}^N \frac{K_i}{N} * 100\%, \quad (22)$$

where N is the top document number and K_i is the amount of the documents which are appeared in the top- i recommended documents. The higher value the t , the better the recommendation is. The result of t vs. λ is shown in Figure 12. The experimental result indicates that the weight of the λ is 0.336 when the best of t is achieved at 0.469.

The tag similarity matrices will be then calculated based on the optimized weight of λ . After the steps described in the above sections, the user profile and document profile will be updated by the tag neighbors, in order to calculate the similarity between them. If the recommended documents are matched the really preferred documents by the candidate user, the recommendation precision will increase.

We average the whole precision for all of the users, and compare the recommendations from top 1 to 40 documents to the users in the experiments. Other two comparable approaches used in the experiments are pure tag vector approach and naive tag neighbor expansion approach. We denote these three approaches as TagNeighborhood with Clustering, TagNeighborhood, Collaborative Filtering and Pure Tag Vector in both of the datasets. In the “MovieLens” dataset, the average precision of TagNeighborhood with Clustering is 84.1, the TagNeighborhood is 78.3, the pure tag vector is 37.4. In the “M-Eco system” dataset, the average precision of TagNeighborhood with Clustering is 57.1, the TagNeighborhood is 50.3, and the pure tag vector is 23.5. The precision comparison of three approaches for top 40 recommendations is shown in Figure 13. In summary, the experimental results validate the advantage of the tag neighbor expansion with clustering in recommendations.

5.3 Personalized Tag Cloud Evaluation

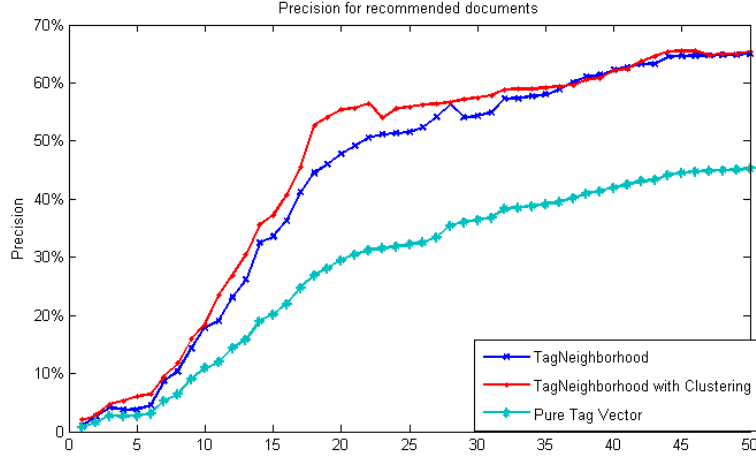


Figure 13: Precision for conditional recommended documents in “M-Eco system”.

Tag Cloud Evaluation for E.coli Outbreak in Germany. The tag cloud model is introduced and motivated in Section 2.6 then technical details are presented in Section 4.4.5. The following section investigates users’ experiences about tag clouds by conducting two experiments. The goal of first experiment was to assess relevancy and utility of parametrized tag clouds. The evaluation was conducted with the group recommendations component 5.5 where the aim of the experiment was to simulate EHEC outbreak in Germany. News recommendations were presented to users through Twitter interface where each suggested item contained a link to the parametrized tag cloud. There are two types of recommendations:

- *Recommended Signal* - such recommended item contains a link to the tag cloud parametrized with a corresponding signal identifier and with a given date; and,
- *Recommended Document* - each recommended document contains an extracted location of the document and a link to the tag cloud parametrized with a corresponding location identifier and with a given date.

For the assessment, the participants in the experiment were asked to assess a relevance of the corresponding tag clouds for the recommended items in Twitter interface. The rating scale was between 1 (dislike) and 5 (excellent) for the relevancy of a given tag cloud. This rating covers the following aspects:

- Is a given tag cloud’s structure appropriate?
- Are depicted tags relevant and meaningful?
- What is the general impression from a given tag cloud?

The participants were not overloaded with more questions as the experiment was lasting for almost a week. For each round participants assessed and rated five different tag clouds. We have obtained 149 user ratings of tag clouds relevance. The average rating was 3.244 and

ratings with evolving time were increasing. The main reason of average relevancy of generated tag clouds was that tags depicted in the tag cloud link to the documents which are related to a given location but their content is too general. We asked participants for additional comments and most of them considered tag clouds as useful retrieval interface which provides a general picture about a given set of documents. However, there are still some drawbacks that should be removed in the future development. Users have complained about not easily recognizable colors of depicted tags.

Another issue concerns the tag syntax in the cloud once some of them were not properly spelled and therefore did not convey the intended meaning e.g., *docto* should be *doctor*. Moreover, a few tags were not meaningful terms e.g., *lyjenmdp*, *lylddg* as they were generated by the automatic generative model of taggings and this issue should be addressed in the future. Furthermore, we have recorded all users' clicks performed during the exploitation of tag clouds. Collected data indicates that a tag cloud is very useful tool for a query refinement as the majority of users started their exploitation by selecting some terms from the tag cloud and then were removing already selected tags.

To illustrate such query refinement one participant has started the exploitation by selecting tags *verwirrung* and *ehc* then he removed term *verwirrung*. Finally, he extended a tags selection by adding a tag *niedersachsen* these resulted into a subset of documents that were annotated with the tags *ehc* and *niedersachsen*. Hence, a tag cloud provides a way how to retrieve only a specific subset of documents from the whole collection.

Users were more active in using tag clouds from the beginning of the experiment as during the last rounds they were assessing a tag cloud relevance by only observing its structure. This limits the detection of user usage patterns of tag clouds. However, the average activity during the initial rounds was five actions per a tag cloud. It includes a tag selection, deletion and also a selection of tags that were assigned to already retrieved documents. Generally, users were exploiting tag clouds as a query refinement tool for retrieving specific subsets of documents related to *E.coli*. Majority of users was attracted to select tags from the following domains:

- medical terms e.g., *koma*, *ehc*, *darminfektion*.
- location e.g., *Spain*, *Niedersachsen*, *Berlin*.
- general terms e.g., *cucumber*, *bauern*, *verwirrung*.
- tags with no descriptive value or no sense e.g., *alle*, *klrduq*.

As was expected, users were retrieving mainly documents based on medical terms and locations. However, even non medical tags (e.g. Spain, cucumber) related to the *E.coli* outbreak were selected by users. These tags were not known by the users in advance and they came up later with additional documents. In a few cases, participants were interested in a selection of meaningless tags. The main advantage of tag clouds in comparison to an ordinary keyword search is the ability to emerge such terms that are not familiar to users in advance. This provides additional context, views which can facilitate a validation process of signals. Participants consider tag clouds as an useful retrieval tool however, the identified drawbacks should be solved in future development.

Personalized Tag Cloud Feedback. The second experiment was conducted with 6 medical experts to assess the usability of our personalized tag cloud. This preliminary evaluation scenario assumes a hypothetical medical expert that created a signal definition matching "cholera"

as a medical condition and “Haiti” as a location. The set of documents belonging to signals that match this signal definition is provided. The participants are then asked to find out more about this outbreak based on the documents found through the interface of the tag cloud provided. Once the exploration process is finished, the participants are asked to answer a set of open-ended sub-questions that correspond to the following evaluation questions:

- Does the tag cloud improves a navigation process through documents?
- Does the structure of the tag cloud correspond sufficiently to the defined signal definition?
- Are the documents retrieved through the tag cloud an appropriate match to the signal definition?

As a result, participants found the personalized tag cloud as useful tool that can improve browsing and retrieving of documents within M-Eco system. A query refinement mechanism clearly improves tag-based information retrieval. Tag cloud should contain more semantically useful tags and not relevant tags should be not displayed. Users preferred a single language tag cloud, therefore it should be avoided of mixing tags with different languages. The complete setup and further details about this user study can be found in the *Section 3.3.3. of M-Eco Deliverable D2.2* [2].

5.4 Evaluation of Generative Model of User Taggings

The goal of the generative model is to assign tags to documents when no or few tagging is available. As mentioned previously, low tagging activity makes difficult to build user profiles and, as a result, it generates inaccurate recommendations. The proposed model (see Section 4.4.6) was evaluated in order to verify the quality of generated taggings. The evaluation was conducted in a such way that users considered generated tags and removed those that were not relevant for given documents. Each participant was asked to evaluate 10 different documents and to assess generated tag assignments by removing not relevant tags (such tags that do not describe or reflect the content of a document).

In total there were 6 different participants, all of them are medical or surveillance experts. Each participant was assessing a different set of documents in order to evaluate all types of documents and make a sample of evaluated documents as representative as possible i.e. evaluated documents should represent a whole collection of available documents in the M-Eco system. The generative model was evaluated by computing precision for each evaluated document. Precision is a common metric in the area of information retrieval which was computed in the following way:

$$Precision(doc_i) = \frac{(|tags_{doc_i}| - |not\ relevant\ tags_{doc_i}|)}{|tags_{doc_i}|}, \quad (23)$$

where $|tags_{doc_i}|$ represents a number of tags of the document doc_i and $|not\ relevant\ tags|$ is the number of not relevant tags stated by the participant. Precision of evaluated documents ranges between 0.5 and 1. The average precision of all evaluated documents is **0.72**. This means that **72%** of generated tags by the generative model were considered by users as relevant. Such precision is considered as acceptable and the generative model replaces an expensive user taggings in a sufficient way. The majority of not relevant tags belongs to the following groups:

- General terms that do not have any descriptive value for a given document e.g., articles, bathroom, buddy, component and country.
- Not meaningful abbreviations that do not have any descriptive value for a given document e.g., dvo, gbm, gdhg, lyhmkbdd, oct., ughh and ucv.
- Names that do not have any descriptive value for a given document e.g., Albert, Hitler, Lauren and Lucy.
- Terms that express some time period and do not have any descriptive value for a given document e.g., days, month, century and time.

Because of the above mentioned problems, the model should be improved to avoid a generation of tags that do not have any descriptive value. Another drawback of this method is a limitation of generating only tags that occur as terms in the given document. To address this problem the model should be extended to produce also tags that are relevant for given documents but do not have to occur within the text of documents. This can be achieved by integrating some medical ontologies or by querying a relevant search engine with generated tags and extract new relevant terms from the retrieved documents. Such extracted terms could be used as new tags of considered documents.

5.5 User Evaluation of Group Recommendations

Group recommendations are discussed in this deliverable in three steps. First we introduce the scenario under which we consider such recommendations (see Section 2.5). The scenario utilizes the EHEC outbreak in Germany that started in May 2011. We then present the group model in Section 4.2.2. This model infers the preferences of individual user actions with published messages and aggregates them to build the group preferences. Finally, the built group preferences are used to produced the recommendations used to publish new messages as described in Section 4.4.4.

In this section, we present two evaluations of group recommendations. The first one, published in [36], aimed at comparing different preference elicitation strategies with our own method, as described in Section 4.2.2. In this evaluation, we used a more open scenario of article recommendations. We show that our approach, called as GroupMender, obtained an average group rating of 3.58 out of 5 over the recommendations given to the group. This represents an improvement of approximately 25% over the best performing strategy we tested. The second evaluation aims at verifying the usefulness of our approach to medical surveillance officials in particular. For that, we use the scenario of the EHEC outbreak in Germany from Section 2.5. We show that the results were similar to the ones achieved on the first evaluation, thus corroborating that group recommendations can also be used within the medical surveillance domain.

Evaluating different group elicitation strategies. In the first evaluation of group recommendations, we compare how different preference elicitation strategies from the literature impact the ratings given by the group of users to the recommendations. We present the different strategies we used in Section 7.3. The research questions of this evaluation were:

1. *How does our approach perform against recommendations generated by approaches based on other group preference elicitation strategies?*

2. *How do the factors used in our model to elicit the preferences influence the overall satisfaction of the group?*

To answer these questions we used Twitter as our test environment as earlier introduced in Sections 2.5 and 4.1. Twitter is a microblogging environment where users are able to publish messages limited to 140 characters (called tweets) and to follow microblog accounts of their preference. A user following an account is able to see all the messages published in that account and a published message is accessible to all followers of the account. That is, a tweet cannot be sent to a specific user, only to the group of followers. Each user also has access to a *timeline* where he can see all the messages from all the users he is following at the moment in a chronological order. A user wishing to quickly share information with their followers can *retweet*, i.e. forward tweets from his timeline to his followers. Users can also mark a message in their timeline as favorite.

To test our group recommendation approach (GroupMender) on Twitter and compare it with others, we created eight Twitter accounts called *gmctrl1* to *gmctrl8*, one for each approach we tested. Besides our group recommendation system, we tested the random, utilitarian (two utility measures - average and multiplicative), plurality voting, least misery, most pleasure, and dictatorship approaches as described in Section 7.3. Due to the number of strategies under evaluation, a controlled and exhaustive experiment was preferred over a long term observation. Twenty users from the Department of Computer Science at Aalborg University took part voluntarily in the experiment. We wanted to have users with similar backgrounds to resemble the scenario of small groups of users interested in the same set of news articles. We also wanted to avoid having synthetic groups. By obtaining the feedback of actual users, we show results that reflect their experience with the platform, instead of hypothetical measures of group satisfaction.

We asked the users to follow all accounts and informed them that these accounts would publish tweets containing links to news articles but not how such publications would take place. To reduce bias from previously published tweets, we conducted one round of publications for each account per day, during five days. All participants were familiarized with Twitter, which facilitated the evaluation. Before the start of the experiment, a practice round was realized to introduce the participants into the experiment, and to assess the initial preferences of the groups and reduce the cold start problem. In this round, for each account we published the same set of 5 randomly selected messages with links from the CNN RSS news feed. The same feed was later used in the actual experiment but the messages published varied according to the strategy of each account. In the particular case of the dictatorship strategy, this round was also used to select the dictator based on the user that interacted the most with the messages published. We decided to choose only this one news feed from CNN to have a similar and congruent set of news articles from the same source when testing the different strategies.

Once the experiment started, every day at the same time the system published a set of five tweets in each account according to their respective strategy. The system ensured that news articles previously published were not republished. This entire procedure was repeated to each group strategy assessed. Each user, then, had to evaluate 40 tweets per day and 200 in total. Number of rounds and messages published by round were chosen to be the minimum necessary to produce statistically significant results.

Figure 19 shows an example of a published tweet. The users were first instructed to read the content of the news article following its link. They were then requested to *rate* each tweet in a scale from 1 to 5, where 1 means a *strong dislike* of the news article the tweet is referring to, and 5 means *strong like*. Ratings are not a feature maintained by Twitter but we built

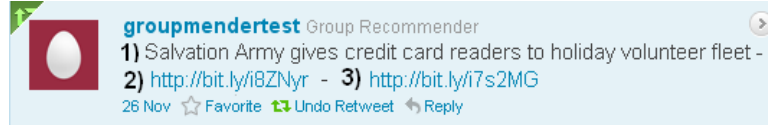


Figure 14: An example of recommended tweet that contains 1) the title of the news article, 2) the link to its contents, and 3) the link to the page where the users could rate the tweet.

a web system provided with a user-friendly interface where the participants could rate each tweet posted. Ratings were requested in order to get a clear picture of the user preferences with respect to the tweets. They were also necessary for the approaches based on the group strategies we compared because they require explicit ratings to determine the group preference.

Besides the rating, the users were instructed to take action on Twitter (i.e. retweet or mark it as favorite) only if they wanted to. If no action was taken on a particular Tweet, it was considered as ignored. Respectively, the independent variables monitored in order to measure the group satisfaction in our approach were: a) the number of retweets (RT), b) the number of tweets marked as favorite (FV) and c) the amount of messages ignored (IM) by all users in a group. The other approaches used the explicit ratings given to the messages as a measure of satisfaction. It is important to note that the participants in this evaluation had already Twitter accounts and a base of followers of their own. We used this information as an assumption that their actions on Twitter during the experiment reflected their actual usage pattern because any misbehavior could potentially be seen by their followers.

To evaluate the results, we computed the total number of tweets marked as favorite, retweeted and ignored by the group represented in each Twitter account per round. We compared these numbers with the average rating of the published tweets in each account per round. In the particular case of our approach, we also looked into the age of the news articles as a measure of number of days from the most recent article published, and compared it with the average ratings given to them.

Figure 15 shows the boxplot with the ratings of the different strategies. Figure 16 shows the ratings over each of the five rounds and an overall value for each strategy. From the boxplot we can notice that the lower ratings of our strategy are almost higher than the higher ratings of all other strategies. Moreover, our strategy had better ratings across all five rounds, the second round receiving the highest. Meanwhile, most of the other approaches had similar results, with average ratings varying between 2.68 and 2.85. The exception was the random strategy which received the worst ratings in general and across all rounds.

Finally, Table 5 shows the overall ratings of each group strategy together with the independent variables RT, FV, and IM. Note that our approach had higher tweets marked as favorite and retweeted while lower number of messages ignored.

We also looked at the date of the recommended articles with our approach in number of days from the most recent article, defined as its age. For example, the oldest article recommended was 20 days older than the most recent one. We then compared the age of the articles with their group ratings, as Figure 17 shows.

There is an inverse correlation between these two factors ($r = -0.57$, $p < 0.01$), meaning that older articles had in general lower ratings. This relationship could be in part attributed to the time decay parameter introduced in our model but also to the higher interest of the group in more recent events. As the group was showing more interest in recent events after each round,

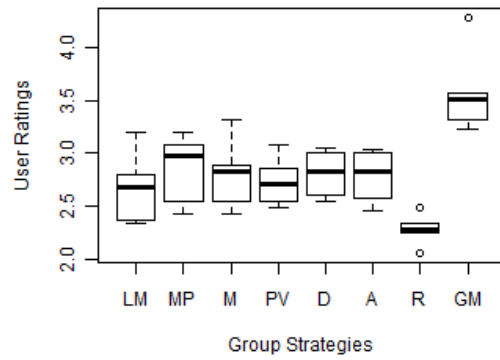


Figure 15: From left to right, boxplot for Least Misery (LM), Most Pleasure (MP), Multiplicative (M), Plurality Voting (PV), Dictatorship (D), Average (A), Random (R), and GroupMender (GM) strategies.

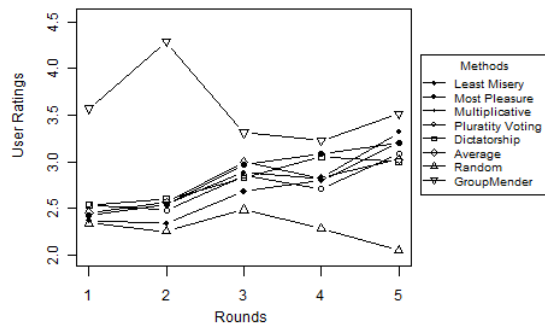


Figure 16: Average ratings by strategy per round.

	Avg. Rating	Favorite	Retweet	Ignored
Random	2.29	14	10	164
Least Misery	2.68	8	25	143
Plurality voting	2.74	12	27	138
Average	2.78	13	19	144
Multiplicative	2.80	15	26	138
Dictatorship	2.81	13	25	139
Most Pleasure	2.85	10	27	153
GroupMender	3.58	19	33	128

Table 5: Average ratings, and number of favorites, retweets and ignored tweets by strategy across all rounds.

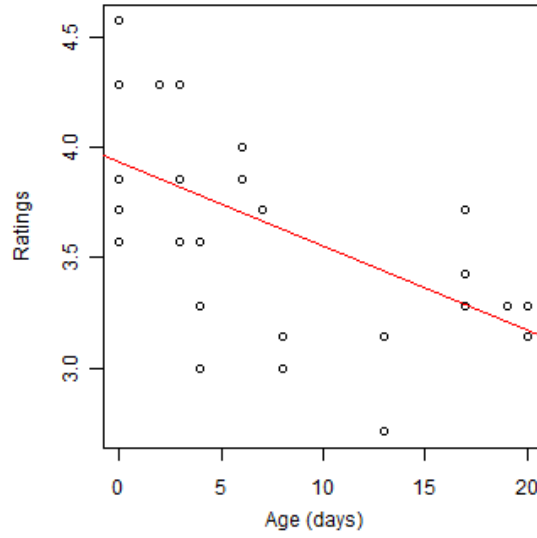


Figure 17: Avg. group ratings by article according to their age.

the system prioritized recommendations related to them, as opposed to older news that the users liked in the previous rounds.

Evaluating group recommendation usefulness to health surveillance officials. In the second evaluation, we perform a similar experiment to the one described above. The main differences are:

- We restrict our users to health surveillance officials only, in order to test group recommendations in the M-Eco system.
- We use the scenario described in Section 2.3 on the EHEC outbreak in Germany.
- We consider the number of clicks in the links of the messages as an additional measure of preference.
- We consider recommendation of locations as described in Section 4.4.3, in order to test

how well users with preferences of locations in Germany receive recommendations related to them.

- We connect the system to tag clouds of items related to the ones published. The tag cloud model was discussed in Section 4.4.5 and its specific evaluation is present in Section 5.3.

In this evaluation we use Twitter again as described above in the previous evaluation and in Sections 2.5 and 4.1. However, this time we do not compare our system with other elicitation strategies. Only our approach, the GroupMender, is tested with health surveillance officials and compared with the results of the previous evaluation. We originally asked 5 of them to follow one Twitter account used to publish the recommended messages⁶. These users were representatives of RKI and NLGA, organizations members of the M-Eco consortium.

Once all of them followed the GroupMender account, we introduced the scenario of the EHEC outbreak in Germany. Specifically, we explained that the actual dates during which the outbreak was detected and progressed would be simulated. For instance, in the first round of recommendations only those items from the first four days after the outbreak was recognized by M-Eco, from May 21 to 25, 2011, would be considered for publication on the Twitter account. Every subsequent round would then consider messages also for the next four days and so on. In total, 7 rounds were realized, one per day, where in the 7th round items from May 21 to June 16th, 2011, were considered for publication.

In each round 5 items were published based on the group preferences elicited from previous rounds. These preferences were aggregated from the individual interactions of the users with the items published. For each item published on Twitter as a message, users could mark them as *favorite*, *retweet* (i.e. forward tweets from his timeline to his followers), or *click* the link in the message to read the full content. Users could also click on a second link in the message, which was connected to a tag cloud of related items. The experience of the users with the tag clouds was evaluated in Section 5.3.

In this evaluation, the tag cloud user interface (GUI) was used also to ask the user for an explicit rating of the item recommended on Twitter. This was implemented in the same way done in the previous evaluation where users were also asked to rate each message explicitly from 1 to 5. Since such ratings are not a feature maintained by Twitter, we had to, exceptionally for this evaluation, incorporate them to the tag cloud GUI. These ratings are then used in this evaluation to compare its results with the previous one.

Figure xx shows the average ratings given by the users per round in this evaluation, together with the results from GroupMender in the first evaluation presented previously. The first round of this evaluation had a high average rating of the items followed by a significant drop. Two explanations for this are possible. One, the first round had round items published because there was no items before to build the group preferences. The high ratings, in this case, could be just a "lucky" set of initial messages.

In addition, the first round had a very limited number of items as the first round accounted only for the first messages of the EHEC outbreak. It is likely then that all the few messages available were significant and any of them to be published would yield high average ratings. This could also explain the low average of the second round. The second round was the first with a larger amount of messages to be considered. Therefore, the initial set of published items in the first round, might had been insufficient to produce good recommendations in the subsequent round. This is even more apparent when noting that all subsequent rounds had much better average ratings than the second.

⁶The account is still open and available at <https://twitter.com/mgroupmender>.

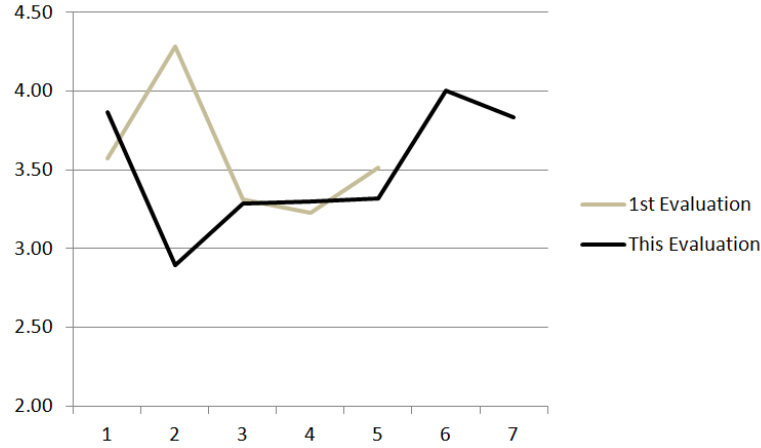


Figure 18: Average ratings of GroupMender per round in the first and in this evaluation.

When comparing this evaluation with the previous, we note that both overall averages across rounds (3.58 in the first and 3.50 in the second) were similar. Results in rounds 3, 4 and 5 also showed similar averages. However, this second evaluation had 2 further rounds that cannot be compared. They highlight a further increase in the average ratings from previous rounds, following a growth pattern. This could be attributed to the recommendation of locations as we show in Section 5.6, because, after each round, the locations became more specific to the interests of the group of users.

The results of this evaluation compared with the previous showed much better results in terms of items marked as favorite or retweeted. In the first evaluation, after 5 rounds, 19 messages were marked as favorite and 33 were retweeted. In this evaluation, after 7 rounds, 49 messages were marked as favorite and 111 were retweeted. While we believe that the messages published were of interest to the group, another reason also might partially explain the higher number of interactions with the messages in this evaluation.

In the first evaluation, the users that took part in the experiment were already users of Twitter and had accounts created with the service before the start of it. These users then were more familiarized with the features of Twitter and the consequences of their actions in the service. By contrast, in the second experiment most users were new to Twitter. Although the features were explained, it is possible that they acted upon each message published by GroupMender as if evaluating the message, and not interacting with as a user of Twitter would. Still, the growth in the number of messages clicked, marked as favorite or retweeted after each round suggests that the recommendations were significant and useful to the users.

5.6 Evaluation of Location Recommendations in a Group Setting

The evaluation of location recommendations was conducted together with the second evaluation of group recommendations, discussed in Section 5.5. The user location preference model was discussed in Section 4.2.3. It basically builds on the locations the user stated in his or her signal definitions together with their corresponding similar locations. The location similarity algorithm together with the recommendation model is discussed in Section 4.4.3. The location



Figure 19: An example of recommended tweet with the location Niedersachsen (Lower Saxony, in German) highlighted in the text.

recommendation model is then used in the group recommendation component (GroupMender) to consider the group location preferences when generating recommendations.

In this section, we briefly show how the items recommended to the group became, after each round, more related to the locations. In the second evaluation of GroupMender, we conducted 7 rounds of recommendations where, in each, 5 new items were published to the Twitter account followed by the group ⁷. Members of RKI and NLGA formed the group being tested. In each round they interacted with the messages published, for example, marking them as favorite or retweeting them. In each message, the location was highlighted as seen in Figure 19.

After all rounds have taken place, we analyzed which locations were recommended. Initially, there were 247 distinct locations from the items that could potentially be published in the first round. Five were randomly published in the first round: USA, Bayern, Berlin and Germany. Note that locations can include cities, regions or states, countries, and even continents. If the group shows an interest in Germany, however, the system will consider locations within Germany such as its cities, according to the political distance described in Section 4.4.3.

In the second round, the locations were Europe, the state of Nevada in the US, Manchester and Germany again. This variation in the second round might also help explain the poor ratings the items obtained in this round, as shown in Figure 18 from Section 5.5. However, from the third round onwards, the locations started to become more specific. They matched the interest of the group for locations in Germany in general (as is generally the interest of RKI) and, more specifically, Lower Saxony (as is the interest of NLGA). The only exception was Spain, present in rounds 4 and 5. The publication of these items can be attributed to the attention received by Spain because it was originally blamed for the origin of the outbreak. Table 6 shows the locations of the items that were published per round.

Next section shows the stress test of the proposed personalization models and how they behave under concurrent conditions.

5.7 Stress tests

The majority of services described in this deliverable presents results to end-users through web interfaces. As an immediate requirement, such services should be able to handle a concurrent access of end-users to the deployed web pages. The motivation is to provide seamless functionality of all services when a large number of users access pertaining web interfaces. It requires to optimize the application so that all possible performance bottlenecks are minimized. During the development process we have identified as the most problematic bottleneck a database access. Therefore, we have integrated a pooling manager that significantly improves the performance of

⁷Twitter was the platform we chose as discussed in Sections 2.5, 4.1 and 5.5

Round	Location	Quantity
1	USA	1
1	Bayern	1
1	Berlin	1
1	Germany	2
2	Europe	1
2	Nevada	1
2	Manchester	1
2	Germany	1
3	Frankfurt	1
3	Lower Saxony	1
3	North Rhine-Westphalia	1
3	Germany	2
4	Spain	2
4	Germany	3
5	Lower Saxony	1
5	Spain	1
5	Germany	3
6	Magdeburg	1
6	Lower Saxony	1
6	Germany	3
7	Lower Saxony	2
7	Germany	3

Table 6: Locations of the items published on GroupMender per round.

the application when there is a need to establish more concurrent connections to the database. Other possible bottlenecks are the JBoss Application Server ⁸ and utilized web framework JBoss Seam ⁹. These underlying applications can significantly slow down the performance if they are not correctly tuned. Performance tuning for the application server JBoss and web framework Seam is conditioned mainly by the utilized hardware devices and it requires individual tuning. However, we list some general tips that can improve the performance of developed services:

- Allocate enough virtual memory for the application server.
- Disable all logging statements that were utilized during the development and testing phase.
- Enable caching of processed data.

The optimization process of the application should be validated with load tests. In this deliverable, we test the developed services with Apache JMeter ¹⁰ - an open source application that provides load functional testing and measures performance. The developed services are subject of load testing such that a specific number of virtual users concurrently accesses the web page of a particular service. To define the testing scenario, there is a need to record appropriate actions using an Internet browser that connects to the proxy server executed by JMeter. Such recorded scenario is automatically imported into JMeter application.

⁸<http://www.jboss.org/jbossas>

⁹<http://seamframework.org/>

¹⁰<http://jmeter.apache.org/>

5.7.1 Tag cloud stress tests

In the following paragraphs, we describe the load testing of a tag cloud service which is introduced and motivated in Section 2.6 and technical details are presented in Section 4.4.5. Firstly, we present a test scenario which simulates a common user behavior. The tag cloud service is tested with different number of concurrent accesses of virtual users. In the end, we present and discuss the results of the stress tests.

The testing scenario tries to simulate a user exploration of the tag cloud. Once, the tag cloud is loaded a user selects a particular tag and a system returns related documents for the given tag. We assume that a user selects more tags in a sequence in order to refine the query. User may also delete some selected tags that he considers as not useful anymore. Tags selection can be extended by clicking on tags of retrieved documents. Based on the above-mentioned assumed typical behavior of users - we design a testing scenario in the following way:

1. Retrieve a tag cloud web page for a given signal (ID 791 related to E.coli outbreak).
2. Call the web service which returns a tag cloud structure for a given signal.
3. Load documents that are matching to tags from the tag cloud.
4. User selects a tag from the tag cloud (verwirrung).
5. User selects another tag from the tag cloud (fordert).
6. User removes already selected tag (fordert).
7. User selects another tag from the tag cloud (health).
8. User selects another tag from the retrieved documents (smallpox).

This testing scenario represents a user utilization of the tag cloud service. The tag cloud service is tested with different number of virtual users which concurrently access and utilize the tag cloud service according to the testing scenario. We observe how the increase of a number of virtual users that are involved in this load testing affects the overall performance of the tag cloud service.

Number of users	Initial loading(sec)	Tag Cloud webser-vice(sec)	Select tag (sec)	Remove selected tag (sec)
1 users	5.74	0.020	13.140	0.313
10 users	6.714	0.021	19.364	0.322
20 users	9.667	0.034	24.042	0.792
30 users	13.381	0.030	35.048	0.824
40 users	31.986	0.040	38.431	1.322
50 users	57.210	0.073	47.145	2.406

Table 7: Tag cloud stress tests results.

The results are presented in the Table 7. The performance of the tag cloud service is acceptable for 20 or less users. However, as more users concurrently access the tag cloud service a site responsiveness drops. When 50 users explore the service concurrently an execution

time for initial loading of documents and tags selection is significantly decreased which is not feasible into a production deployment of this service. These two operations are the mostly time consuming because they fetch a large number of documents from the database. The reason why we are not reaching ideal performance concerns the current deployment conditions. The server where a tag cloud service is currently hosted connects to the database which is hosted on another just ordinary machine and this causes a bottleneck. To avoid this issue in the production deployment a database should be hosted on a robust database server or even on the same server as the application of tag cloud service.

Worth mentioning the tag cloud service was running during these stress tests without any application errors or exceptions. The implementation was optimized to correctly handle a connection pooling to the database and only limitation during the tests was available hardware.

6 Deployment

In this section we introduce the two ways of accessing our personalization components, either through a stand alone application or using our web services.

6.1 Installation Requirements

In order to run the components as a stand alone application the following software requirements must be fulfilled:

- Java SDK 6;
- Eclipse Java EE edition;
- Jboss 4.2.3 or later from <http://www.jboss.org/jbossas/downloads.html>;
- Apache Ant (it should come with eclipse);
- Subversion plug-in for Eclipse @ <http://subversion.tigris.org>;
- Checkout the project @ <https://iwis.svn.sourceforge.net/svnroot/iwis/WP5>;
- SQL Server with Microsoft SQL Server Management Studio version 10.0.1600.22 or later. This temporary solution because we are changing to PostgreSQL 8.3 due to license constraints.

Configuration and Deployment. After software installation, some configuration must take place:

- Edit your environment variables JBOSS-HOME, ANT-HOME and JAVA-HOME. These names for the variables are suggestive;
- Set your "JBOSS-HOME" in the "build.properties" file and 'seam-gen.properties' file;
- The database configuration should be set in the class database. *IDatabaseConstants.java* and WP5-dev-ds.xml;
- To deploy the system, simply run the task "build.xml" in the target deploy;

- The system should be accessible from the URL `http://localhost:8080/WP5/home.seam`. There you will find a link to the applications.

Data loading. This deployment guideline does not ensure the data loading, which should be gathered primarily by web services of the system. However, for applications that are going to access location-based recommendations there is a requirement of loading the GeoNames database ¹¹. The article Loading GeoNames Data Into SQL Server 2008 provides a detailed guide line on how to load GeoNames data into SQL Server 2008.

Calling Components Methods. The personalization services are primarily intended to be accessed through our web services (see Section 6.2), however our components can be directly called through the following methods:

- Class: *SignalDB*, Method: *getSignalsByUser(userId, startDate, endDate)* - it returns a signals relevant for a given user ID within a period of time;
- Class: *TagCloudService*, Method: *getTagCloud(signalId)* - it returns a list of tags for a given signal ID;
- Class: *IndicatorDB*, Method: *getRecommendedIndicators(userId)* - it returns a list of recommended indicators for a given user ID;
- Class: *DocumentDB*, Method: *getDocumentsByIndicatorId(indicatorId)* - it returns a list of documents for a given indicator ID.

6.2 Web Services

The major goal of the WP5 web services is to support integration between the project partners, and provide outsiders with personalized information processed by our components. In order to reach such an integration, we implemented web services providers and consumers. The web services providers are interface where personalization services such as recommendations are available for external access. The web services consumes are client applications that gather information provided by others web services, usually from other partners.

We publish our personalization services as REST Web Services, to which external applications only need to make a HTTP request to access the desired service. The REST Web Service call was implemented to return objects either in XML or in JSON format. The web service is called using the basic URL path:

`http://{server:port}/{workpackage}/{institution}/rest/{serviceName},`

e.g.,

`http://localhost:8080/WP5/aau/rest/recommendations`

¹¹<http://www.geonames.org/>

6.2.1 Web Service Providers

In the following we present the web services which provide personalized information.

- **getUserSignalList()** - it returns a list of user signals that were already matched to one of his/her signal definitions in a given period(start date, end date). No documents go attached to the signal list. <http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/userSignalNoDocuments/{userId}/{startDate}/{endDate}/>
- **getListRecommendedSignalList()** - it returns list of recommended signals for a given user in a given period (start date, end date). <http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/recommendedSignalList/{userId}/{startDate}/{endDate}/>
- **getUserSignal()** - it returns a list of user signals along with documents that were already matched to one of his/her signal definitions in a given period (start date, end date). <http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/userSignalWithDocuments/{userId}/{startDate}/{endDate}/>
- **tagCloudBySignal()** - it returns a tag cloud as JSON object for a given signal identifier. <http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/tagCloudBySignal/{signalId}/>
- **tagCloudByLocation()** - it returns a tag cloud for all tags assigned to a set of all documents for a given location identifier. The service returns a tag cloud structure encoded as JSON object. The set of considered documents can be restricted with a given time interval. <http://demos.iwis.cs.aau.dk:8081/WP5/aau/rest/services/tagCloudByLocation/{locationId}/{startDate}/{endDate}/>

6.2.2 Web Service Consumers

For the matter of integration with other parties that provide information, we also implemented web services clients that consume information. They are:

- **getDocumentList()** - it returns a list of documents extracted from Twitter by WP3;
- **getDocument ()** - it returns a single document by its ID extracted from Twitter by WP3;
- **getSourceList()** - it returns a list of sources from the where the documents are retrieved.

7 Related work

This section reviews and compare a number of related works that contributed to the comparison against our own models. In each related work of this section, we compare technical details, identify differences and outline our improvements over baseline approaches.

7.1 User Modeling

The user information exploited for building the user and group profile of our methods were originally extracted from social media applications (e.g., Twitter) where users voluntarily share information of interest such as symptoms or announce eventual local epidemy, etc. Traditional

bio-surveillance algorithms for public health count the number of occurrences of one or more observation variables in a given spatial location over a given time period to look for anomalous patterns [10]. This is typically done on structured and formal sources of information (such as the number of reported patients suffering from influenza). For example, these algorithms take as input a training-window, and upper control limit, which may be represented in terms of the number of standard deviations from an expected (mean) behavior [29]. A model of expected behavior is built from the past window of the time series data to make a prediction about the present. Such a prediction takes the form of a triple consisting of: set of observation variables, a window of time, and a boolean flag indicating whether the behavior of observed variables, exceeds an expected behavior, or the given time window. [8] consider profiles that can be augmented as dynamic, which means they can change over time. For example, in [56] the user profile is incrementally updated over time, accumulating his past preferences. Similarly, [54] propose an approach to learn what the authors called an ontological user profile, and to incrementally update it, also addressing changing preferences of the users over time. In [47] recommender systems are discussed in the context of collaborative filtering algorithms. The authors argue that time decay measures can be used to weight up the relationship of items that were rated in a short time frame instead of far apart. Similarly as previous approaches, our group recommendation model (see Section 4.4.4) assume that the interactions that took place more recently represent a higher indicator of the current preferences of the group.

7.2 Recommendations for Health Care

Recommendations aims at the natural process of humans to make decisions among various possibility available. For the M-Eco project, our recommendation method were developed with the aim at helping medical experts in the analysis of signals outbreaks. Similarly, [49] design a system based on Semantic Web Technologies to develop a system with the capability to assist health care professionals regarding the possible medication or drug to prescribe, according to some selection criteria. Unlike [49], user’s preferences are not considered, this is a key difference from our methods given that our methods in general include a personalization factor in their models. [14] present a method for automated generation of medical recommendations, using the combined power of the topic maps and expert systems. To obtain new knowledge from topic maps and to integrate this knowledge with a medical decision making systems, keywords for interrogation are needed. The keywords are extracted from various sources of medical decision making systems: patient data documents, rules documents or even recommendations.

Similarly as [49], [14]’ work neglects user’s preference while generating recommendations. [28] propose a trust-based recommender system for patients who are confronted with the decision of which physician to trust. The proposed system codifies privacy concerns in a privacy-friendly framework and present two architectures that realize it: the Secure Processing Architecture (SPA) and the Anonymous Contributions Architecture (ACA). In SPA, patients submit their ratings in a protected form without revealing any information about their data, and the computation of recommendations proceeds over the protected data using secure multi-party computation techniques. In ACA, patients submit their ratings in the clear, but no link between a submission and patient data can be made. The paper discusses various aspects of both architectures including techniques for ensuring reliability of computed recommendations and system performance, and provide their comparison. The difference from our methods concerns with the fact that our model does not rely on any trust-based model. Instead, in our work the recommendations take into account individual user interest in contents (e.g., diseases or

epidemics) from the health expert point of view.

7.3 Group Recommendations

Most of related works to our recommendations model are not designed for supporting groups of experts dealing with epidemic detection. Instead, they explore different strategies on how to gather and model the group preference. [52] classifies them into three categories: majority-based, consensus-based, and borderline strategies. The majority-based strategies use the most popular items (or item categories) among group members. For instance, GroupCast [43] displays content that suits the intersection of user profiles when the persons are close to a public screen. The consensus-based strategies consider the preferences of all group members. As an example, MusicFX [42] recommends the most relevant music station in a fitness center using a group profile computed by summing the squared individual preferences. The borderline strategies consider only a subset of items, in individual profiles, based on user roles or other relevant criteria. For example, PolyLens [44] uses the Least Misery strategy to recommend movies for small user groups based on the MovieLens database.

These strategies, however, typically require explicit individual ratings so that users can indicate different gradations of preference [52]. These explicit ratings are used in different preference elicitation strategies that aggregate individual preferences. [41] and [40] discuss such strategies of which we considered for comparison in our evaluation discussed in Section 5.5:

- *Plurality Voting* – each member votes for his preferred item (or item category) and the one with the highest votes is selected. Then, this method is reiterated on the remaining items (item categories) in order to obtain a ranked list;
- *Utilitarian Strategy* – utility values of each item are calculated by some pre-defined function. This can be done, for example, utilizing the *average* of the preferences of all the group members or in a *multiplicative* calculus where the ratings of all group members for one item are multiplied;
- *Dictatorship Strategy* – uses the preferences of only one member, who imposes his tastes to the rest of the group. If possible to identify, this member is typically the one considered the most respected by the others;
- *Least Misery Strategy* – considers for each item the rating with the minimum value. Then, the item with the highest minimum value is assumed to be the preferred one by the group;
- *Most Pleasure Strategy* – considers for each item the rating with the maximum value. Then, the item with the highest maximum value is assumed to be the preferred one by the group.

One of the examples of systems using one of those strategies is PolyLens, where the explicit ratings the individual users give to the movies are used. In addition to the problem of relying on explicit user ratings, their proposed system did not account for changing user preferences over time. Similarly, [33] proposes a travel system where users need to explicitly rate the different aspects involved in a travel plan (e.g., hotel facilities or leisure activities). On the other hand, [52] does attempt to infer user preferences. The authors use TV viewing data to compare different group recommendation strategies, given a reference profile. This is done by eliciting individual preferences from the time each individual spent watching each movie,

i.e. the more a person watches a movie, the more it is assumed that he is interested in it. However, their work is outside the scope of web systems and has a different focus, resulting in incomparable results. While they propose an approach to try to determine the factors that would help choose a group recommendation strategy, we propose a new strategy to attempt to generate recommendations giving an existing group, regardless of any previously existing parameter or reference. In addition to that, our group recommendation model utilizes a time decay factor in order to privilege more recent news that can possibly determine more accurately the group's interest.

7.4 Tag Expansion in Recommendation

Tags have been recently studied in the context of recommender systems due to various reasons. Recommendations of relevant documents should be based on the sufficient occurrences for similar signals expressed by tags. We review the related literatures from the perspectives of tag expansion and tag clustering. K. R. Bayyapu and P. Dolog in [5] try to solve the problems of sparse data and low quality of tags from related domains. They suggest using tag neighbors for tag expression expansion. However the tag neighbors are based on the content of documents. We propose another approach to extend the tag set for the user profile by collaborative filtering approach. [61] proposes a collaborative filtering approach TBCF (Tagbased Collaborative Filtering) based on the semantic distance among tags assigned by different users. That is, two users could be considered similar not only if they rated the items similarly, but also if they have similar understanding over these items. To calculate the semantic similarity, the WordNet dictionary is being accessed to find the shortest path connecting a tag and its synonym in the graph synsets. In [9], an interesting approach was proposed to model the documents in social tagging systems as a document graph. The relevance of tag propagated along edges of the documents graph is determined via a scoring scheme, with which the tag prediction was carried out. Heymann et al. [27] addressed the same problem of tag prediction based on the anchor text, web page content and the surrounding hosts. A binary classifier was trained on a set of very popular bookmarks to differentiate the closest tags. [21, 53] demonstrated how tag clusters serving as coherent topics can aid in the social recommendation of search and navigation. In [24] topic relevant partitions are created by clustering documents rather than tags. By clustering of documents, it improves recommendation by distinguishing between alternative meanings of a query. While in [11], clusters of documents are shown to improve recommendation by categorizing the documents into topic domains.

7.5 Personalized Tag cloud

Sinclair et al. [55] considers tag clouds as useful retrieval interface when a user's searching task is not specific. Users by exploration of tag clouds get familiar with a domain of the system. According to [48] a tag cloud allows users to perform 4 different tasks: i) search - retrieving matching content to the selected term in the tag cloud, ii) browsing - user can browse available documents, not necessarily to search for some particular topic or task, iii) impression formation - user gains an impression which topics are dominant for the documents associated with the tag cloud and iv) recognition - user can recognize which of different documents a tag cloud is more likely to visualize. This retrieval interface also supports a query refinement during the search task as by addition or deletion of tags a placed search query changes. This strategy was considered our tag neighbor expansion of user queries as shown in Section 4.3.3.

In medical surveillance systems there is a need to assess a huge number of documents in a short time. A surveillance personnel has to search, browse and create an impression from the explored documents. Therefore, a tag cloud is suitable retrieval interface which can improve a validation process of documents. The majority of tag clouds visualize tags in alphabetical order however, there is a lot of ongoing research about depicting tags in some semantical manners. [23, 15] propose to group semantically related tags and depict them in a tag cloud near by with similar color. Such approach provides better orientation in the tag cloud as related tags can be easier identified by users. Tags are clustered based on their co-occurrences. Similarly, the proposed tag cloud generation method also groups semantically similar tags. However, our approach is more robust as syntactically similar tags are firstly pre-clustered (grouping singular, plural or misspellings of tags) similarly as proposed in [57]. It leads to tag space reduction as resulted tag cloud does not contain syntactically similar tags. In the second phase, tags are similarly clustered based on tags co-occurrences but our proposed approach also considers retrieved semantical distances from WordNet dictionary if available. A folksonomy contains a huge number of tags therefore there is a need to select only the most important that will be depicted in the tag cloud. [58] proposes different tags selection algorithms and our method utilizes a similar selection technique where tags with higher coverage are preferred. However, our method selects tags with higher coverage from different clusters. It results into a semantically more diverse tags thus a user can explore and browse more topics from the generated tag cloud.

In comparison to all related work the proposed method can generate parameterized tag clouds based on different parameters such as location, time, signal which allows to restrict set of considered documents and in consequence also tags. Such Moreover, a tag cloud can be personalized if user profile is available.

7.6 Spatial Reasoning

Spatial reasoning has been constantly studied in the literature in applications there orientation is the primary value driver. Spatial reasoning encompasses of two main abilities: the ability of calling up images in mind and the ability to reason with these images. Spatial Reasoning typically applies to Geographic Information Systems (GIS) and often assumes that exact coordinates are known and inferences can be carried out to make a decision. In computer science, lots of effort have been put on *collision detection* to tell whether two objects occupy the same space at the same time and *path planning* to plan the best trajectories so that two objects avoid collisions [17, 35].

Our location-based method (see Section 4.4.3) presents the model for user location preferences, which is built from the set of user locations defined in his or her signal definitions and from similar locations. In particular, the location similarity is computed as a function of two factors such as *political hierarchy* (e.g., city, state or country level) and *distance and population*. To the best of our knowledge, almost no report in the literature performs spatial reasoning considering the factors addressed in our model. For this reason, we will discuss a number of techniques and compare them without our approach at the technical level regardless the medical domain. [20] present a new approach to represent qualitative spatial knowledge and to spatial reasoning. This cognitive considerations motivates the approach and is based on relative orientation information about spatial environments. The approach aims at exploiting properties of physical space which surface when the spatial knowledge is structured according to conceptual neighborhood of spatial relations. In our method, the notion of conceptual neigh-

neighborhood is also considered once the outbreak warning are target to cities nearby the outbreak focus. In line with our method [20] proposes a formal method for qualitative reasoning about distances and cardinal directions in geographic space. The main problem addressed is how to infer the distance and direction from point two points in a space. Our method differs from [20] in the sense that we look into the neighborhood location by considering additional factors to determine the minimal distance between two points.

The exposure of user data in social web applications has attracted attention of researchers that try to estimate the location of web content or people on the web based on the analysis of geo-related terms. Some applications focused on extracting explicitly geographic information from web pages such as address or points of interest [1, 38]. [12] propose and evaluate a probabilistic framework for estimating a Twitter user’s city-level location based purely on the content of the user’s tweets, even in the absence of any other geospatial cues. The framework relies on three key aspects i) tweet content, ii) a classification component for automatically identifying words in tweets with a strong local geo-scope; and iii) a lattice-based neighborhood smoothing model for refining a user’s location estimate. The framework aims at estimating k possible locations for each user in descending order of confidence. Our model converges with [12]’ study in the sense that we also explore Tweeter data for predicting neighborhood locations, however we are not focused on finding people. In addition our method. Crandall et al. [13] investigate how to organize a large collection of geotagged photos. They try to combine textual and visual features to place images on a map. They have restrictions in their task that their system focuses on which of ten landmarks in a given city is the scope of an image. As to methods that support social surveillance, [51] investigates the detection of earthquakes with real-time Twitter data. In order to make such a prediction, they make use of location information for tracking the how of information across time and space. For predicting earthquake, their algorithm needs to maintain a knowledge base of where and when the earthquake is reported. Our work also location-based method also relies on a knowledge base but a medical one.

8 Conclusion and Future Works

In this report we presented the WP5 personalization components implemented for the M-Eco project addressing diverse aspects such as motivation scenarios, formal models, design and the requirements necessary to install and realize the benefits of personalization in M-Eco.

As shown, most of personalization in M-Eco is conceived by observing user’s tagging activity aiming at building tag-based user models and thereby delivering the appropriate information to right users. We leverage user tagging information to personalize recommendations, group users and personalize tag cloud. In addition to that, we reason on top the geolocation information in order to relax recommendations and reach users in the surround the focus of an eventual epidemy. Furthermore, we make the personalization services available on the Web through the implementation of REST web services.

As a future work, we plan to provide justifications which explain why recommendations were generated. We believe that this feature could make the system more reliable and improves the user interaction in the system. From the user’s interface perspective, we plan to increase the user’s interactivity with the system. Users could for instance change manually the ranking of search results or recommendations. Another goal is to integrate hierarchical tag clouds which will enable users to explore more tags from tag clouds and as consequence to perform narrower query refinement search. Further, users could drag and drop widgets to arrange the interface

they prefer to work. These future plans are based on realistic analysis on what could bring value to the M-Eco system in terms of personalization.

References

- [1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM, 2004.
- [2] G. Backfried, E. Diaz-Aviles, P. Dolog, J. Dreesman, F. Durao, T. Eckmanns, G. Kirchner, M. Krieck, R. G. Lage, M. Leginus, L. Otrusina, A. Stewart, and E. Velasco. D2.2 evaluation report. Technical report, August 2011. EU FP7 Small or Medium-scale Focused Research Projects (STREP).
- [3] G. Backfried, E. Diaz-Aviles, J. Dreesman, T. Eckmanns, G. Kirchner, M. Krieck, R. G. Lage, Lubomir, A. Stewart, and E. Velasco. Requirements specification and evaluation guidelines for the event-based surveillance system (d2.1). Technical report, August 2010. EU FP7 Small or Medium-scale Focused Research Projects (STREP).
- [4] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamas, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information retrieval*, 13(3):291–314, 2010.
- [5] K. R. Bayyapu and P. Dolog. Tag and Neighbour Based Recommender System for Medical Events. In *Proceedings of MEDEX 2010: The First International Workshop on Web Science and Information Exchange in the Medical Web co-located with WWW 2010 conference*, 2010.
- [6] L. Boratto and S. Carta. State-of-the-art in group recommendation and new approaches for automatic identification of groups. In A. Soro, E. Vargiu, G. Armano, and G. Paddeu, editors, *Information Retrieval and Mining in Distributed Environments*, volume 324 of *Studies in Computational Intelligence*, pages 1–20. Springer Berlin / Heidelberg, 2011.
- [7] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, March 2001.
- [8] P. Brusilovsky, A. Kobsa, and W. Nejdl, editors. *The Adaptive Web*, volume 4321. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [9] A. Budura, S. Michel, P. Cudré-Mauroux, and K. Aberer. Neighborhood-based tag prediction. *The Semantic Web: Research and Applications*, pages 608–622, 2009.
- [10] W. W. Chapman. Natural language processing for biosurveillance. 1(3), 2005.
- [11] H. Chen and S. Dumais. Bringing order to the web: automatically categorizing search results. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, New York, NY, USA, 2000. ACM.

- [12] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM.
- [13] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.
- [14] D. Dragu, V. Gomoï, and V. Stoicu-Tivadar. Automatic generation of medical recommendations using topic maps as knowledge source. In *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, pages 593–597, may 2011.
- [15] F. Durao, P. Dolog, M. Leginus, and R. Lage. Simspectrum: A similarity based spectral clustering approach to generate a tag cloud. In *Proceedings of (accepted for) the 11th International Conference on Web Engineering*, 2011.
- [16] F. Echarte, J. Astrain, A. Córdoba, and J. Villadangos. Pattern matching techniques to identify syntactic variations of tags in folksonomies. *Emerging Technologies and Information Systems for the Knowledge Society*, pages 557–564, 2008.
- [17] K. Forbus, J. Mahoney, and K. Dill. How qualitative spatial reasoning can improve strategy game ais. *Intelligent Systems, IEEE*, 17(4):25–30, 2002.
- [18] C. Frank, M. Faber, M. Askar, H. Bernard, A. Fruth, A. Gilsdorf, M. Höhle, H. Karch, G. Krause, R. Prager, et al. Large and ongoing outbreak of haemolytic uraemic syndrome, germany, may 2011. *Euro Surveill*, 16:21, 2011.
- [19] C. Frank, D. Werber, J. Cramer, M. Askar, M. Faber, M. Heiden, H. Bernard, A. Fruth, R. Prager, A. Spode, et al. Epidemic profile of shiga-toxin-producing escherichia coli o104: H4 outbreak in germany preliminary report. *New England Journal of Medicine*, 2011.
- [20] C. Freksa. Using orientation information for qualitative spatial reasoning. *Theories and methods of spatio-temporal reasoning in geographic space*, pages 162–178, 1992.
- [21] J. Gemmell, A. Shepitsen, M. Mobasher, and R. Burke. Personalization in folksonomies based on tag clustering. In *Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, July 2008.
- [22] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies*, pages 25–28. Citeseer, 2006.
- [23] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *MERIDA INSCIT2006 CONFERENCE*, 2006.
- [24] C. Hayes and P. Avesani. Using tags and clustering to identify topic-relevant blogs. In *International Conference on Weblogs and Social Media*, March 2007.
- [25] M. Hearst. *Search user interfaces*. Cambridge Univ Pr, 2009.

- [26] M.-E. Hernandez, S. M. Falconer, M.-A. Storey, S. Carini, and I. Sim. Synchronized tag clouds for exploring semi-structured clinical trial data. In *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*, CASCON '08, pages 4:42–4:56, New York, NY, USA, 2008. ACM.
- [27] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.
- [28] T. R. Hoens, M. Blanton, and N. V. Chawla. Reliable medical recommendation systems with patient privacy. In *Proceedings of the 1st ACM International Health Informatics Symposium*, IHI '10, pages 173–182, New York, NY, USA, 2010. ACM.
- [29] M. Höhle. surveillance: An r package for the surveillance of infectious diseases. 2007.
- [30] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, chapter 31, pages 411–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [31] A. Huang. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008)*, Christchurch, New Zealand, pages 49–56, 2008.
- [32] R. K. Institute. Ehec/hus o104:h4 outbreak. Technical report, May 2011. Germany. Technical report.
- [33] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 48–54, New York, NY, USA, 2004. ACM.
- [34] A. M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59–68, Jan. 2009.
- [35] L. Kotovsky and R. Baillargeon. The development of calibration-based reasoning about collision events in young infants. *Cognition*, 67(3):311–351, 1998.
- [36] R. Lage, F. Durao, and P. Dolog. Towards effective group recommendations for microblogging users. In *SAC '12: Proceedings of the 29th ACM Symposium on Applied Computing*, New York, NY, USA, 2012. ACM.
- [37] R. Lage, F. Durao, P. Dolog, and A. Stewart. Applicability of recommender systems to medical surveillance systems. In *The Second International Workshop on Web Science and Information Exchange in the Medical Web (MedEX 2011) co-located with the 20th ACM Conference on Information and Knowledge Management*, 2011.
- [38] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

- [39] M. Leginus and V. Zemaitis. Speeding up tensor based recommenders with clustered tag space and improving quality of recommendations with non-negative tensor factorization. Master's thesis, Aalborg University, 2011.
- [40] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.
- [41] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, 16(3-4):281–319, 2006.
- [42] J. F. McCarthy and T. D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372, New York, NY, USA, 1998. ACM.
- [43] J. F. McCarthy, T. J. Costa, and E. S. Liongosari. Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 332–345, London, UK, 2001. Springer-Verlag.
- [44] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: a recommender system for groups of users. In *ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers.
- [45] L. Otrusina, P. Smrz, and G. Backfried. Speech recognition and content classification subsystems. Technical report, December 2010. EU FP7 Small or Medium-scale Focused Research Projects (STREP).
- [46] D. Ramage, P. Heymann, C. Manning, and H. Garcia-Molina. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63. ACM, 2009.
- [47] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer US, Boston, MA, 2011.
- [48] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 995–998, New York, NY, USA, 2007. ACM.
- [49] A. Rodríguez, E. Jiménez, J. Fernández, M. Eccius, J. M. Gómez, G. Alor-Hernandez, R. Posada-Gomez, and C. Laufer. Semmed: Applying semantic web to medical recommendation systems. In *Proceedings of the 2009 First International Conference on Intensive Applications and Services*, pages 47–52, Washington, DC, USA, 2009. IEEE Computer Society.
- [50] G. X. Rong Pan and P. Dolog. Improving recommendations in tag-based systems with spectral clustering of tag neighbors. In *The 3rd FTRA International Conference on Computer Science and its Applications (CSA-11)*, 2011.

- [51] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.
- [52] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In P. De Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 40–51. Springer Berlin / Heidelberg, 2010.
- [53] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266, New York, NY, USA, 2008. ACM.
- [54] A. Sieg, B. Mobasher, and R. Burke. Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, pages 39–46, Barcelona, Spain, 2010. ACM. ACM ID: 1869452.
- [55] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *J. Inf. Sci.*, 34:15–29, February 2008.
- [56] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 675–684, New York, NY, USA, 2004. ACM. ACM ID: 988764.
- [57] J. van Dam, D. Vandic, F. Hogenboom, and F. Frasincar. Searching and browsing tag spaces using the semantic tag clustering search framework. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 436–439. IEEE, 2010.
- [58] P. Venetis, G. Koutrika, and H. Garcia-Molina. On the selection of tags for tag clouds. In *Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11*, pages 835–844, New York, NY, USA, 2011. ACM.
- [59] W. Wahlster and A. Kobsa. User models in dialog systems. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin, Heidelberg, 1989.
- [60] G. Xu, K. Bayyapu, R. Pan, R. Lage, and P. Dolog. Personalization and adaptation component for the m-eco system. Technical report, December 2010. EU FP7 Small or Medium-scale Focused Research Projects (STREP).
- [61] S. Zhao, N. Du, A. Nauerz, X. Zhang, Q. Yuan, and R. Fu. Improved recommendation based on collaborative tagging behaviors. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 413–416. ACM, 2008.